H. Bernstein

Mikroprozessor

2

ISBN 3-921682-26-8

Es kann keine Gewähr übernommen werden, daß die in diesem Buche verwendeten Angaben, Schaltungen, Warenbezeichnungen und Warenzeichen frei von Schutzrechten Dritter sind. Alle Angaben werden nur für Amateurzwecke mitgeteilt. Alle Daten und Vergleichsangaben sind als unverbindliche Hinweise zu verstehen. Sie geben in keinem Falle Aufschluß über evtl. Liefermöglichkeiten oder Verfügbarkeiten. In jedem Falle sind die Datenblätter der Hersteller heranzuziehen. Nachdruck und öffentliche Wiedergabe, besonders die Übersetzung in andere Sprachen nur mit vorheriger Genehmigung des Verlages. Irrtum sowie alle Rechte vorbehalten. Beim Betrieb von Sende-und Empfangs-Anlagen sind unbedingt die Bestimmungen der Deutschen Bundespost zu beachten.

Copyright by Ing.W.Hofacker GmbH , 8000 München 75 ,Postfach 75437 . Druck der 1.Auflage im September 1977.

Gedruckt in der Bundesrepublik Deutschland ,printed in West-Germany, Imprimé en RFA

Mikroprozessor

Inhaltsverzeichnis

1.	Technologie der Mikroprozessor-Speicherbausteine	1
1.1	Schreib-Lese-Speicher in TTL-Technik	1
1.2	Statische Schreib-Lese-Speicher in MOS-Technik	21
1.3	Dynamische Schreib-Lese-Speicher in MOS-Technik	30
1.4	Schreib-Lese-Speicher in CMOS-Technik	37
1.5	Festwertspeicher (ROM)	40
1.6	Programmierbare Festwertspeicher (PROM)	58
1.7	Schieberegister	69
2.	Aufbau eines Mikroprozessor-Systems mit dem 8080	81
2.1	Der Mikroprozessor 8080	82
2.2.	Taktgenerator 8224	107
2.3	System-Steuerungs-Baustein 8228	110
2.4	Schnittstellen	112

Vorwort

Dieses Buch beschäftigt sich mit der Hardware der Mikroprozessoren. Hierzu gehören die Speicher-Bausteine und die Bausteine des Mikroprozessor-Systems 8080.

In dem Kapitel der Speicher-Bausteine werden die wichtigsten Speicherelemente gezeigt. Die Schreib-Lese-Speicher in TTL-Technik sind sehr schnelle Speicher, aber die Leistungsaufnahme ist sehr groß. Dagegen sind die Speicher in MOS-Technik langsam, aber die Packungsdichte ist sehr groß. Bei den MOS-Speichern unterscheidet man zwischen den statischen und dynamischen Typen.
Der Schreib-Lese-Speicher in CMOS-Technik zeichnet sich besonders durch geringe Leistungsaufnahme und variabler Betriebsspannung aus. Daher werden CMOS-Speicher in tragbaren Geräten und als "Floppy-Disk"-Einheiten verwendet.

Bei den Festwertspeichern unterscheidet man zwischen den Masken-ROMs, den programmierbaren Speichern und den löschbaren ROMs. Die Masken-ROMs werden in fertigen Systemen verwendet, da sie sehr billig sind. Der Inhalt kann aber direkt programmiert werden. Der gespeicherte Inhalt ist ebenfalls nicht löschbar. Die EPROMs oder REPROMs sind durch UV-Licht löschbar. Nach dem Löschen kann ein neues Programm eingegeben werden.

Bei dem zweiten Teil wird der Mikroprozessor 8080 beschrieben, einschließlich seinen anderen Bausteinen. Zum Schluß dieses Buches finden Sie zwei kleine Mikroprozessor-Systeme, die einfach nachzubauen sind.
Der Autor möchte sich bei folgenden Firmen bedanken:

Siemens, München
Intel, München

München 1977
H. Bernstein

1. Technologie der Mikroprozessor Speicherbausteine

1.1 Schreib-Lese-Speicher in TTL-Technik

Ein RAM (Random Access Memory) ist ein Schreib-Lese-Speicher, bei dem jede einzelne Speicherzelle addressierbar ist. Unter "addressierbar" verstehen wir den Zugriff jeder einzelnen Speicherzelle durch eine Adresse:

Adresse	Speicherplatz
0	0
1	1
2	2

Ist jede Speicherzelle in einem Schreib-Lese-Speicher adressierbar, so sprechen wir von einem volladressierbaren oder volldecodierten RAM. Durch die Adresse selektieren wir den Speicherplatz. Danach schreiben wir eine Information ein oder lesen eine gespeicherte Information aus.

Die einzelnen Schreib-Lese-Speicher unterscheiden sich in den Technologien und in den verschiedenen Wortlängen. Bei den Wortlängen spricht man auch von der Organisation des Speichers, z.B.: 256 x 1, 256 x 4 usw. Mit den 256 kennzeichnen wir die Adressen oder die Datenwörter. Das RAM hat damit 256 Adressen oder Datenwörter. Die Wortlänge dagegen beträgt in dem ersten Beispiel 1 und in dem zweiten 4. Mit der Wortlänge wird gleichzeitig der Ein- und Ausgang festgelegt. Hat ein RAM die Wortlänge von 1 RAM, hat das RAM nur einen Ein/Ausgang. Bei einer Wortlänge von 4 ergeben sich vier Ein/Ausgänge.
Ein RAM mit der Organisation von 256 x 1 hat acht Adressen-Eingänge, da $2^8 = 256$ ist und einen Ein- und Ausgang für die Daten. Das RAM 256 x 4 hat ebenfalls acht Adressen-Eingänge, aber vier Ein- und Ausgänge für die Daten. Damit können wir in ein 1-Bit-RAM immer nur ein Bit einschreiben oder auslesen. Bei einem 4-Bit-RAM dagegen vier Bit parallel einschreiben oder auslesen.

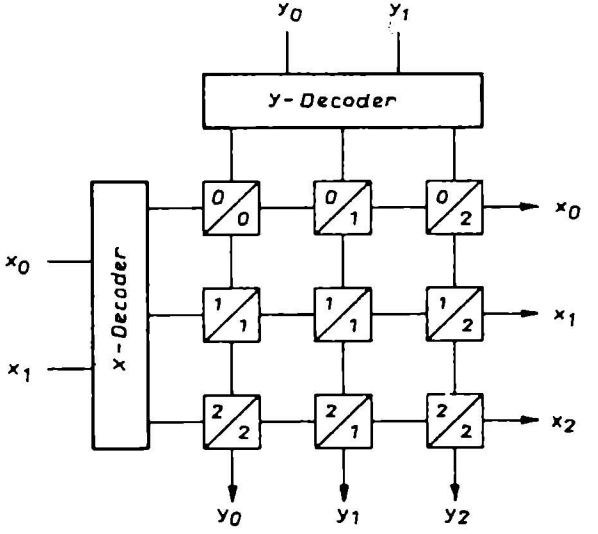


Bild 1.1.1 Aufbau eines Bit-orientierten Schreib-Lese-Speichers (RAM)

In Bild 1.1.1 ist ein Bit-orientiertes-RAM gezeigt. Die Adressierung der einzelnen Speicherzellen erfolgt über einen X- und Y-Decoder. Durch den X-Decoder werden die einzelnen Zeilen (rown) adressiert. Wir sprechen daher auch vom Zeilen-Decoder oder Rown-Decoder. Es ergibt sich für das Bild 1.1.1 folgende Funktionstabelle:

Eingänge		Zeilen
X_1	X_0	
L	L	x_0
L	H	x_1
H	L	x_2

Der Zeilen-Decoder decodiert die X-Adresse und adressiert damit die einzelnen Zeilen in dem Schreib-Lese-Speicher.

Die Selektion der Speicherzellen wird aber nur durch das Zusammenwirken der Zeilen mit den Spalten vorgenommen. Der Y-Decoder steuert die einzelnen Spalten in dem Schreib-Lese-Speicher an. Durch den Y-Decoder werden die einzelnen Spalten (column) adressiert. Wir sprechen daher auch vom Spalten-Decoder oder Column-Decoder. Es ergibt sich für das Bild 1.1.1 folgende Funktionstabelle:

Eingänge		Spalten
Y_1	Y_0	
L	L	y_0
L	H	y_1
H	L	y_2

Setzt man die beiden Funktionstabellen zusammen, so ergibt sich die gesamte Betriebsart für die Adressierung:

Eingänge				Speicherplätze
X_1	X_0	Y_1	Y_0	
L	L	L	L	0/0
L	L	L	H	0/1
L	L	H	L	0/2
L	H	L	L	1/0
L	H	L	H	1/1
L	H	H	L	1/2
H	L	L	L	2/0
H	L	L	H	2/1
H	L	H	L	2/2

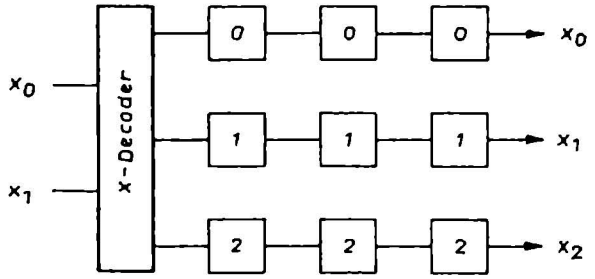


Bild 1.1.2 Aufbau eines Zeilen-orientierten Schreib-Lese-Speichers (RAM)

Durch die Eingänge X und Y können wir jeden Speicherplatz in der Speichermatrix ansteuern, adressieren.
Wesentlich anders ist das Zeilen-orientierte-RAM von Bild 1.1.2 aufgebaut. Das RAM hat nur den X-Decoder für die Zeilenansteuerung. Mit jeder Adresse wird eine Zeile adressiert. Es ergibt sich daher folgende Funktionstabelle:

Eingänge		Zeile
X_1	X_0	
L	L	x_0
L	H	x_1
H	L	x_2

Vergleichen wir Bild 1.1.1 und Bild 1.1.2, so erkennen wir folgendes. Das Bit-orientierte-RAM wird für eine 1-Bit-Organisation, z.B. 64 x 1, 256 x 1 oder 1024 x 1 usw. verwendet. Dagegen kann das Zeilen-orientierte-RAM für eine 4-, 8- oder 16-Bit-Organisation, z.B. 64x4,256x8 oder 1024x16 usw. verwendet werden.
Die Terminologie der Schreib-Lese-Speicher ist immer in der Organisation zu suchen. Aus der Organisation erkennen wir den Aufbau eines Schreib-Lese-Speichers.

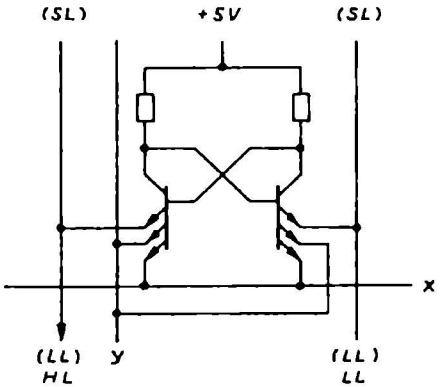


Bild 1.1.3 Aufbau einer Bit-orientierten Speicherzelle

Das Bild 1.1.3 zeigt eine Bit-orientierte-TTL-RAM-Speicherzelle. Die beiden Transistoren der Speicherzelle sind in Multiemitter-Technik aufgebaut, d.h. jeder Transistor in der Speicherzelle hat drei Emitter, aber nur einen Collektor und eine Basis.
Die Speicherzelle von Bild 1.1.3 wird über die x- und y-Leitung adressiert. Das Einschreiben oder das Lesen einer Information erfolgt über die beiden SL/LL-Leitungen. Über die Schreib-Leitung und die Lese-Leitung kann eine Information in eine adressierte Speicherzelle eingeschrieben oder ausgelesen werden. Die beiden Leitungen bezeichnet man daher als Schreib-Lese-Leitungen. Die Information wird durch einen Schreib-Verstärker auf die Schreib-Lese-Leitung gegeben. Während dieses Vorgangs ist der Lese-Verstärker gesperrt. Arbeitet der Lese-Verstärker, ist der Schreib-Verstärker gesperrt. Damit wird der Schreib-Lese-Vorgang getrennt
In Bild 1.1.3 ist immer ein Transistor leitend und der andere dadurch gesperrt. Daher ist in der Speicherzelle entweder ein L- oder ein H-Signal gespeichert. Durch diese Tatsache ist eine der beiden Schreib-Lese-Leitungen die L-Signal-Leitung und die andere die H-Signal-Leitung. Wir können mit der L-Signal-Leitung ein L-Signal in die Speicherzelle einschreiben oder auf der anderen Leitung ein H-Signal. Daher werden die beiden Schreib-Lese-Leitungen auch als:

L-Schreib-Lese-Leitung (LL) und

H-Schreib-Lese-Leitung (HL)

bezeichnet.
Erfolgt nun eine Ansteuerung des X- und Y-Decoders durch eine Adresse, so geben die beiden Decoder auf eine bestimmte Zeile und auf eine bestimmte Spalte ein H-Signal ab. Bei der betroffenen Zelle tritt damit eine Selektion durch die beiden H-Signale auf und die Speicherzelle ist adressiert. In der so adressierten Speicherzelle fließt damit der Collektorstrom des leitenden Transistors in eine der beiden Schreib-Lese-Leitungen. Dadurch erhält die betreffende Leitung ein Signal und der Leseverstärker erkennt ein Signal an. An dem Ausgang des RAMs erscheint die gespeicherte Information.
Für das Einschreiben einer Information gilt die gleiche Adressierung durch den X- und Y-Decoder. Soll ein H-Signal eingeschrieben werden, so geht die H-Schreib-Lese-Leitung auf L-Signal. Soll dagegen ein L-Signal eingeschrieben werden, so schaltet die L-Schreib-Lese-Leitung auf L-Signal. In die Speicherzelle kann nur mit einem L-Signal auf den beiden Schreib-Lese-Leitungen eingeschrieben werden. Die andere Schreib-Lese-Leitung muß auf H-Signal liegen. Das L-Signal auf einer der beiden Leitungen wird nur bei der adressierten Speicherzelle wirksam.
Mit einem L-Signal an der Schreib-Lese-Leitung und einem H-Signal an der x- und y-Leitung wird die Speicherzelle gesetzt. Ist die Speicherzelle aber bereits gesetzt, so bleibt das Schreibsignal unwirksam. Dies ist besonders für den Anwender wichtig, da es hierbei zu keiner Programmänderung kommt, wenn bereits ein Signal eingeschrieben ist. Die Speicherzelle kann nur mit einem L-Signal auf der anderen Schreib-Lese-Leitung verändert werden.

Eine Zeilen-orientierte-TTL-RAM-Speicherzelle ist dagegen anders aufgebaut. Hierzu benötigen wir nur eine x-Zeilen-Leitung für die Adressierung. Die Schreib-Lese-Leitungen sind im wesentlichen gleich von Bild 1.1.3 und 1.1.4. Das RAM

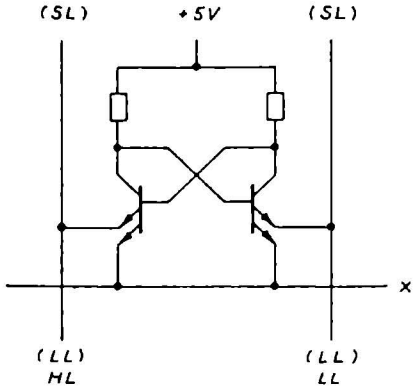


Bild 1.1.4 Aufbau einer Zeilen-orientierten Speicherzelle

von Bild 1.1.4 hat pro Speicherzelle zwei Transistoren in Multiemitter-Technik. Jeder Transistor hat aber nur zwei Emitter, einen Collektor und eine Basis. Die Adressierung verläuft zeilenweise. Bei einer Adressierung des X-Decoders erhält die adressierte Zeile ein H-Signal. Damit liegen die gesamten angeschlossenen Emitter einer Zeile auf einem H-Signal. Mit den einzelnen Schreib-Lese-Leitungen können nun in die betreffende Zeile L- oder H-Signale eingeschrieben werden.
Durch ein L-Signal auf einer Schreib-Lese-Leitung können wir die adressierte Speicherzelle kippen. Hat die Speicherzelle aber diese Information gespeichert, so ändert sie ihren Zustand nicht. Bei dem Kipp- oder Setzvorgang wird der angesteuerte Transistor leitend und der andere durch die kreuzweise Verbindung der Basis mit dem Collektor gesperrt. Beim Lesen kann dagegen der gesamte Collektorstrom auf die Schreib-Lese-Leitung fließen. Das Auslesen einer Information erfolgt zerstörungsfrei.

Das Bild 1.1.5 zeigt den Aufbau eines TTL-RAMs mit den einzelnen Schreib- und Lese-Verstärkern. Über den H-Eingang erfolgt die Ansteuerung der H-Schreib-Lese-Leitung und über den L-Eingang die L-Schreib-Lese-Leitung. Für die beiden Schreib-Lese-Leitungen sind zwei Lese-Verstärker vorhanden. Die Ausgänge der Lese-Verstärker haben offene Collektoren für die Erweiterung der Speicherkapazitäten.

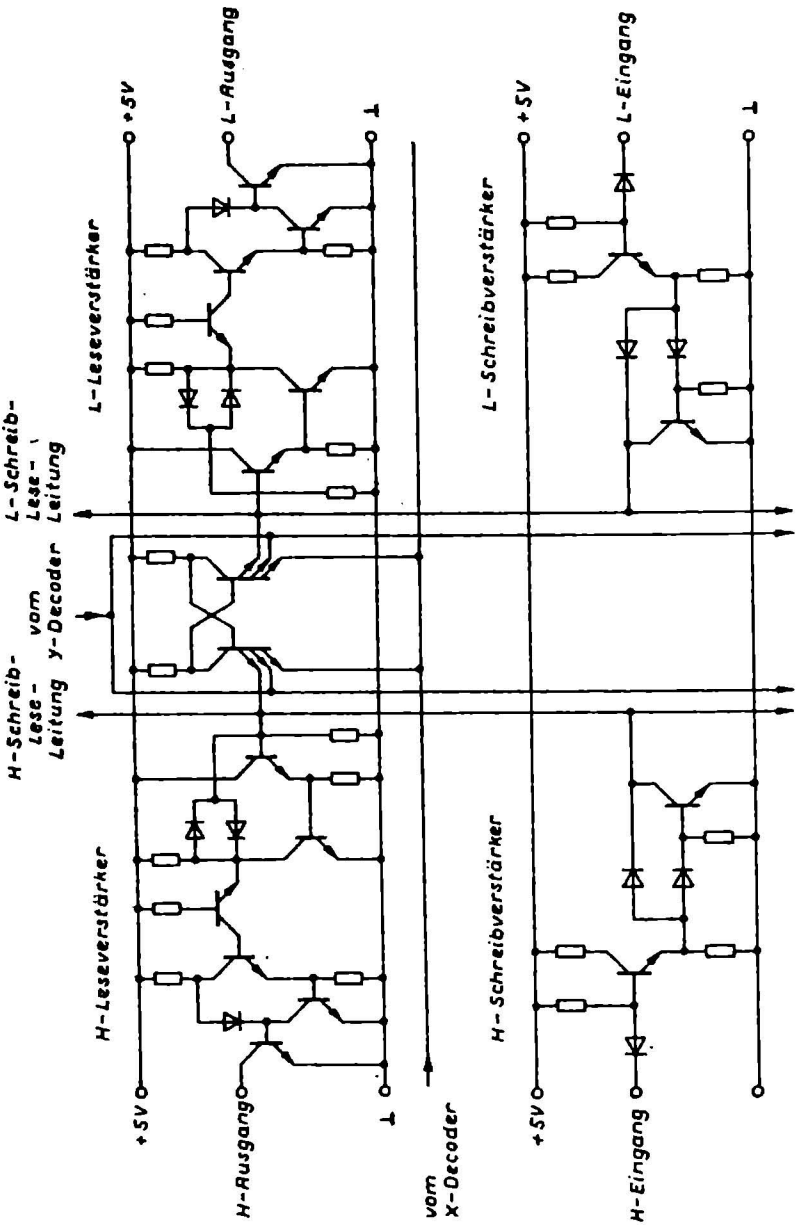


Bild 1.1.5 Aufbau einer Schreib-Lese-Speicherzelle mit Schreib-und Leseverstärker

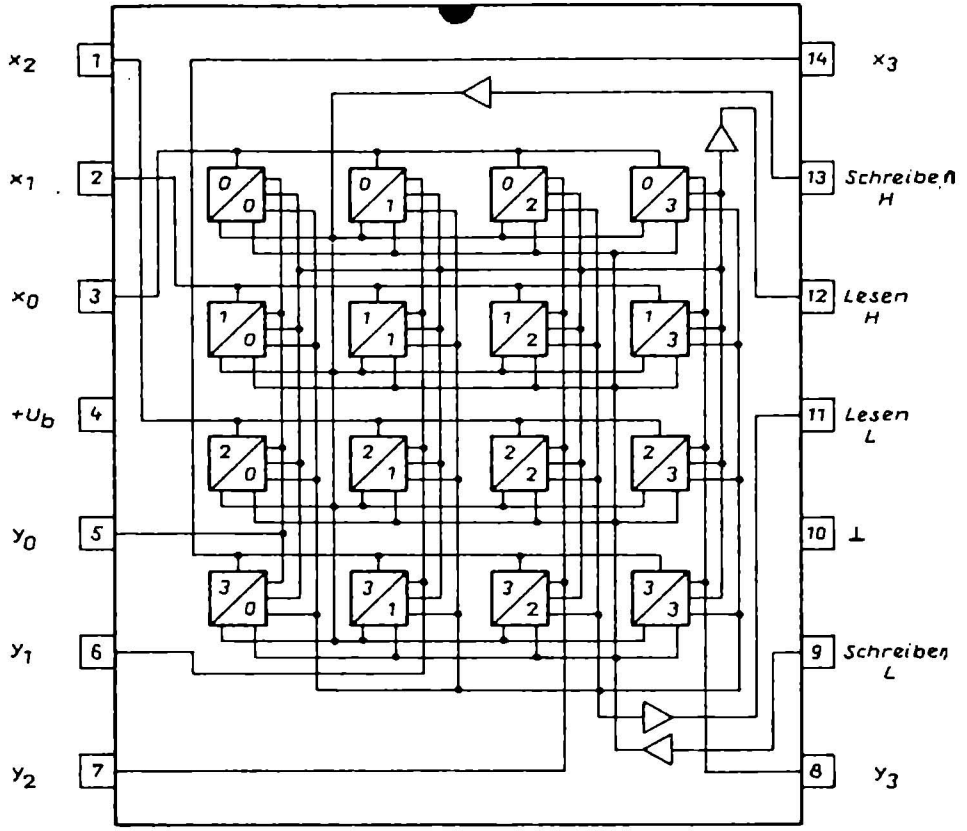


Bild 1.1.6 Anschlußbelegung des TTL-RAMs 7481

In Bild 1.1.6 ist die Anschlußbelegung eines bipolaren 16-Bit-RAMs gezeigt. Es handelt sich hierbei um den Typ 7481. Für dieses RAM wird kein X- und kein Y-Decoder benötigt. Die Ansteuerung erfolgt direkt über die Anschlußbelegung. Es ergibt sich folgende Arbeitsweise:

Adressierung (addressing).

Mit den X-Eingängen wird die betreffende X-Zeile ausgesucht (selektiert). Die Y-Spalte wird von den Y-Eingängen bestimmt. Ein H-Signal auf einem X- und Y-Eingang bedeutet eine Adressierung einer Speicherzelle.

Schreiben (write):

Bei einem Schreib-Vorgang wird der Schreib-Verstärker aktiviert. Nach einer abgeschlossenen Adressierung kann eingeschrieben werden. Bei der Einschreibung eines H-Signals wird der Eingang "Schreiben-H" auf ein H-Signal gelegt. Bei der Einschreibung eines L-Signales wird der Eingang "Schreiben-L" auf ein H-Signal gelegt.

Lesen (read):

Die beiden Schreib-Eingänge sind hierbei auf L-Signal zu legen. Danach kann adressiert werden. Hat die adressierte Speicherzelle ein H-Signal gespeichert, so geht der Ausgang "Lesen-H" auf H-Signal. Ist in der Speicherzelle ein L-Signal gespeichert, so geht der Ausgang "Lesen-L" auf H-Signal.

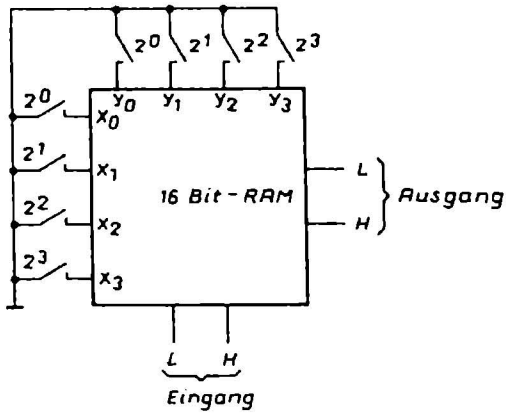


Bild 1.1.7 Testschaltung für das TTL-RAM 7481

Anhand der beiden Ein- und Ausgänge können in das TTL-RAM die anliegenden Informationen eingeschrieben oder gespeicherte Informationen ausgelesen werden. Die ausgelesenen Daten werden dabei nicht zerstört, sondern bleiben erhalten.
In Bild 1.1.7 ist eine Testschaltung für ein 16-Bit-TTL-RAM gezeigt. Die Adressierung der Speicherzellen wird hier über Schalter vorgenommen. Sind die Schalter offen, so liegt an den Eingängen ein H-Signal an. Dieses H-Signal ist durch die TTL-Technik bedingt. Nach einer Adressierung können wir die an den beiden Eingängen anliegenden Daten einschreiben oder die gespeicherten Informationen auslesen.
Wesentlich einfacher ist die Schaltung von Bild 1.1.8. Hierbei wird die Ansteuerung des 16-Bit-RAMs durch die beiden 2-Bit-Vorwahlzähler vorgenommen. Die beiden Vorwahlzähler liegen an einem gemeinsamen Taktsignal an. Durch eine externe Ansteuerung können wir die Vorwahlzähler programmieren. Ab einem bestimmten Zeitpunkt werden die beiden Zähler freigegeben und sie zählen hoch. Ist der vorprogrammierte Zählerstand erreicht, bleibt der betreffende Zähler stehen. Für die Adressierung des 16-Bit-RAMs sind daher maximal vier Taktimpulse erforderlich. Danach kann eingeschrieben oder ausgelesen werden. Zwischen den Vorwahlzählern und dem TTL-RAM müssen Decoder eingeschaltet sein. Dadurch wird der Zählerstand der einzelnen Zähler decodiert.

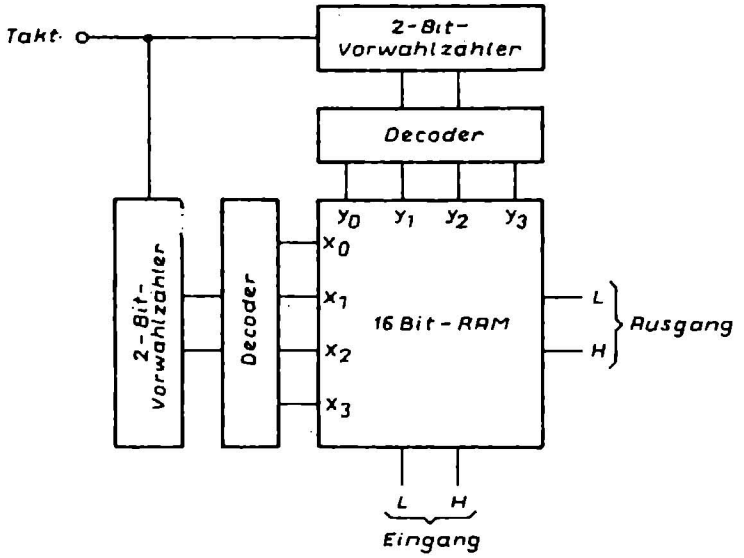


Bild 1.1.8 Ansteuerung des TTL-RAMs 7481 durch zwei Vorwahlzähler

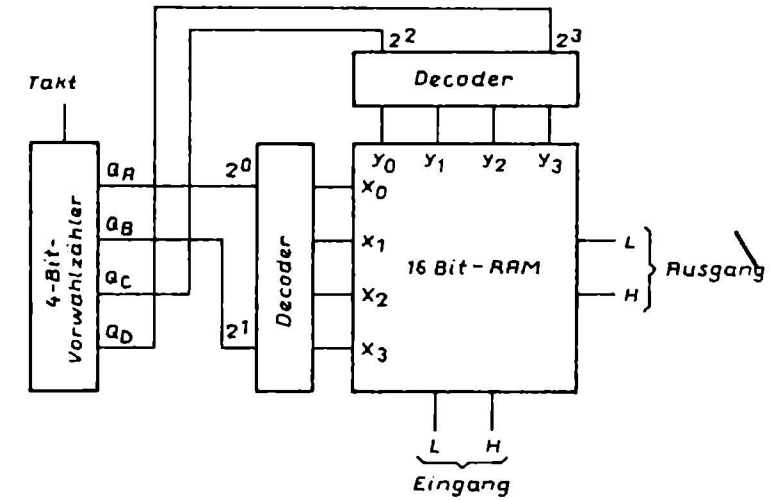


Bild 1.1.9 Ansteuerung des TTL-RAMs durch einen 4-Bit-Zähler

Die Schaltung von Bild 1.1.9 zeigt die gesamte Adressierung durch einen 4-Bit-Vorwahlzähler. Mittels der Vorwahl ergibt sich folgende Adressierung:

Adresse	Zählerstand				Decoder-Ausgänge								Speicherzelle
Nr.	Q_D	Q_C	Q_B	Q_A	Y_3	Y_2	Y_1	Y_0	X_3	X_2	X_1	X_0	X/Y
0	L	L	L	L	L	L	L	H	L	L	L	H	0/0
1	L	L	L	H	L	L	L	H	L	L	H	L	1/0
2	L	L	H	L	L	L	L	H	L	H	L	L	2/0
3	L	L	H	H	L	L	L	H	H	L	L	L	3/0
4	L	H	L	L	L	L	H	L	L	L	L	H	0/1
5	L	H	L	H	L	L	H	L	L	L	H	L	1/1
6	L	H	H	L	L	L	H	L	L	H	L	L	2/1
7	L	H	H	H	L	L	H	L	H	L	L	L	3/1
8	H	L	L	L	L	H	L	L	L	L	L	H	0/2
9	H	L	L	H	L	H	L	L	L	L	H	L	1/2
10	H	L	H	L	L	H	L	L	L	H	L	L	2/2
11	H	L	H	H	L	H	L	L	H	L	L	L	3/2
12	H	H	L	L	H	L	L	L	L	L	L	H	0/3
13	H	H	L	H	H	L	L	L	L	L	H	L	1/3
14	H	H	H	L	H	L	L	L	L	H	L	L	2/3
15	H	H	H	H	H	L	L	L	H	L	L	L	3/3

Mittels der Vorwahlschalter können wir eine der 16 Adressen vorwählen. Nach Abschluß des Wählvorganges geben wir einen Takt auf den Vorwahlzähler. Der Zähler adressiert zwar jede Speicherzelle separat an, aber während dieser Zeit darf nicht eingeschrieben werden. Liegen die beiden Schreib-Eingänge auf L-Signal, so können die Speicherinformationen ausgelesen werden. Ist die Adressierung für den Schreib-Vorgang abgeschlossen, so können wir in die betreffende Speicherzelle eine Information einschreiben.

Vergleichen wir die beiden Schaltungen von Bild 1.1.8 und 1.1.9, ergibt sich in der Art der Adressierung ein wesentlicher Unterschied. Die Adressierung von Bild 1.1.8 kann bedeutend schneller durchgeführt werden. Der Grund ist die separate Adressierung der X- und Y-Eingänge. Bei der Adressierung in Bild 1.1.9 ist der Zähler gezwungen, jeden Speicherplatz separat zu adressieren, wenn er die Speicherzelle 3/3 selektieren muß. Hierzu sind 16 Taktimpulse notwendig. In der Schaltung von Bild 1.1.8 werden hierzu nur vier Taktimpulse benötigt.

Die Adressierungsart von Bild 1.1.8 wird bei dem 8080 Mikroprozessor-System verwendet:

nADR = niederwertige Adresse

oder

low add = low address

Für die Y-Adressierung lauten die Bezeichnungen:

hADR = höherwertige Adresse

oder

hi add = high address

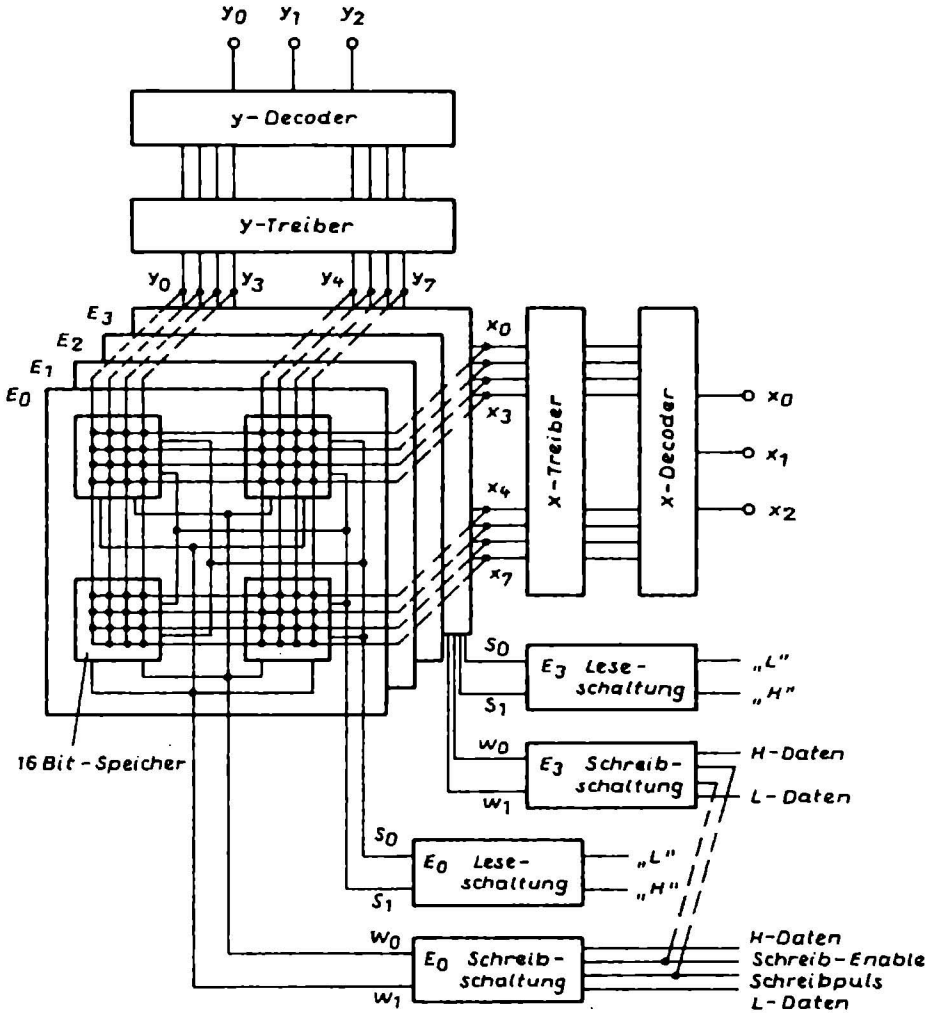


Bild 1.1.10 Aufbau einer 64 x 4-Bit-Speichermatrix

Das Bild 1.1.10 zeigt einen Aufbau einer 64 x 4-Bit-Speichermatrix. Zu der X- und Y-Adressierung kommt in diesem Fall noch die Z- oder E-Adressierung hinzu. Die X- und Y-Adressierung liegt über Decoder an den vier Speicherebenen an. Da wir vier Ebenen hintereinander in der Z-Richtung haben, ergibt sich für uns eine 4-Bit-Wortspeicherung. Wir können 64 Wörter mit je 4 Bits in der Schaltung von Bild 1.1.10 speichern. Durch die drei X-Eingänge erhalten wir decodiert acht Möglichkeiten zur Ansteuerung der Zeilen in der Speichermatrix. Mit den drei Y-Eingängen erhalten wir ebenfalls acht decodierte Signale für die Spalten. Damit können wir durch die 2 x 3-Adressen-Bits 8 x 8 = 64 Wörter adressieren.
Für das Einschreiben und das Auslesen sind vier Schreib- und vier Lese-Schaltungen erforderlich. An allen vier Schreib-Schaltungen liegt der Schreib-Enable und der Schreibimpuls an. An den H- und L-Daten-Eingängen liegen bei den Schreib-Schaltungen die Eingangswörter mit je vier Bits an.

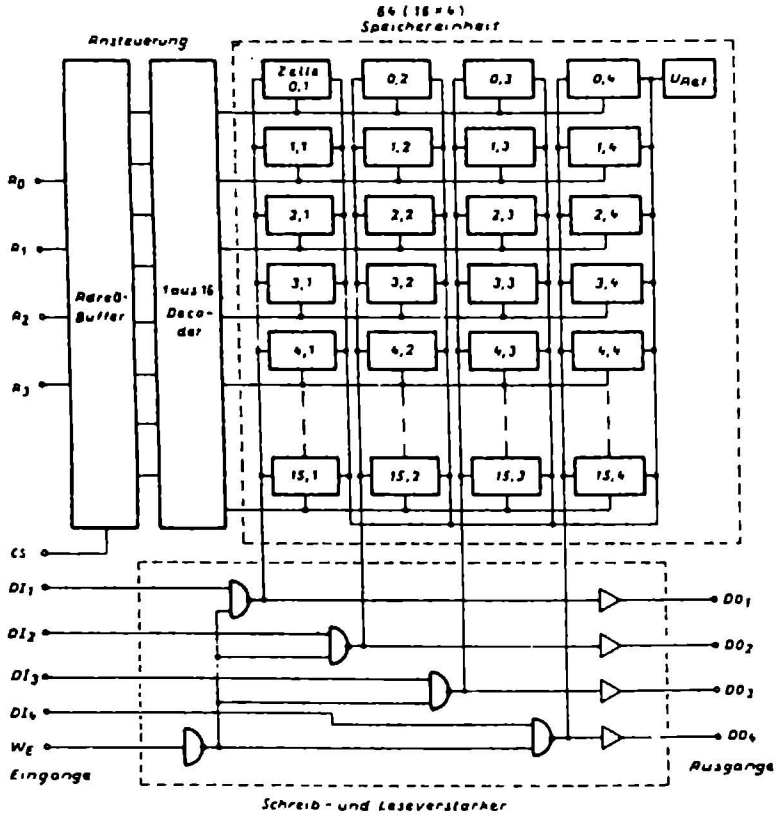


Bild 1.1.11 Aufbau eines 64-Bit-Schreib-Lese-Speichers mit der Organisation von 16 x 4-Bits (TTL-RAM 7489)

Das Bild 1.1.11 zeigt den Aufbau eines 64-Bit-Schreib-Lese-Speichers mit einer Organisation von 16 x 4-Bits. Hierbei handelt es sich um den Baustein 7489. Über dem Adressen-Buffer liegen die vier Eingangs-Adressen an dem 1-aus 16-Decoder. Die 16 Ausgänge steuern die einzelnen Speicherplatz-Zeilen an. Jeder der 16 Zeilen besteht aus vier Speicherplätzen und daher handelt es sich hierbei um ein Zeilen-orientiertes-RAM. Da der 1- aus 16-Decoder jede der 16 Zeilen adressieren kann, sprechen wir von einem volldecodierten RAM. Es können 16 Adressen zu je vier Bits gespeichert werden.

Adressierung:

Die Adressierung wird über die vier Adressen-Eingänge A_0 bis A_3 vorgenommen. Gesperrt wird die Adressierung durch den CS-Eingang (chip select) Die Ansteuerung erfolgt im Dual-Code. Beim Lesen des Speicherinhaltes muß der CS-Eingang auf L-Signal und beim Schreiben auf H-Signal liegen.

Lesen:

Der Speicherinhalt des RAMs kann nur gelesen werden, wenn der CS-Eingang auf L-Signal liegt. Der Schreib-Eingang (write enable) muß während des Lesevorganges auf H-Signal liegen. Die gespeicherten Informationen liegen parallel an den DO_1 bis DO_4 - Ausgängen (data output).

Schreiben:

Während des Schreibens muß der WE-Eingang auf L-Signal liegen. Die zu speichernden Daten liegen parallel an den Dateneingängen DI_1 bis DI_4 an und werden eingeschrieben. Geht der WE-Eingang auf H-Signal, ist die angelegte Information oder das 4-Bit-Wort eingeschrieben.

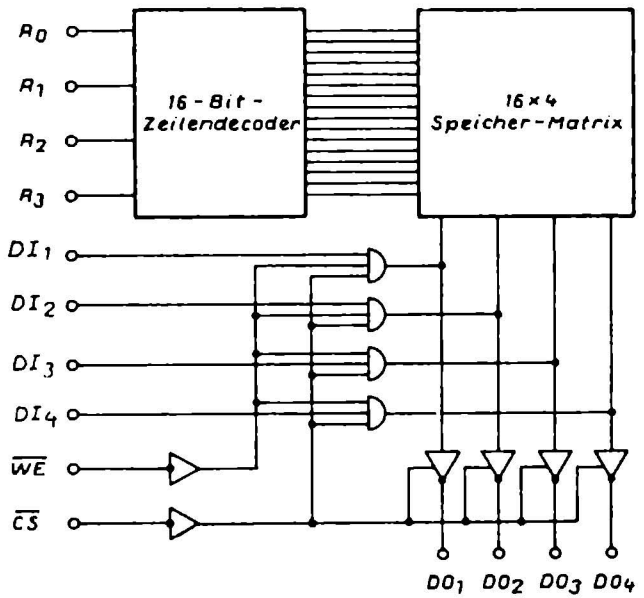


Bild 1.1.12 Vereinfachte Darstellung eines 64-Bit-Schreib-Lese-Speichers vom Typ 7489

Der Schreib-Lese-Speicher gibt seine gespeicherten Informationen invertiert wieder aus. Sollen die Eingangsdaten mit den Ausgangsdaten identisch sein, so müssen hinter dem RAM von Bild 1.1.11 vier NICHT-Gatter geschaltet werden.
In Bild 1.1.12 ist eine andere Darstellung für ein 64-Bit-RAM gezeigt. Die vier Adressen-Eingänge A_0 bis A_3 steuern den Zeilendecoder an. Durch die Decodierung werden aus den vier Adressen-Eingängen 16 Zeilen. Mit den 16 Zeilen erfolgt die Ansteuerung der Speichermatrix.
An den vier Eingängen liegen Informationen an. Das Einschreiben der Informationen hängt von den beiden Eingangssignalen des WE- und CS-Einganges ab. Bei einem L-Signal an den beiden Eingängen kann eine Information eingeschrieben werden. Das eingeschriebene Datenwort liegt bereits während des Einschreibens an den Ausgängen invertiert an.
Die Ausgänge werden durch den CS-Eingang gesteuert. Hat der CS-Eingang ein H-Signal, können keine Informationen eingeschrieben oder ausgelesen werden. Die Ausgangsstufe für die vier Ausgänge ist gesperrt. Die TTL-Technik bietet dem Anwender drei Möglichkeiten für die Ausgangsstufen.
Bei den TTL-RAMs gibt es drei verschiedene Ausgangsschaltungen und zwar die Gegentakt- oder Totempol-Ausgänge, die Eintakt- oder Open-Collektor- oder "oben-ohne-Ausgänge" und die Tri-State oder Three-State-Ausgänge.

Gegentakt-Ausgänge:

Diese Ausgänge haben eine B-Gegentakt-Endstufe. Bild 1.1.13 zeigt die Innenschaltung für diesen Ausgang. Wir haben zwei npn-Ausgangstransistoren. Die Ansteuerung der beiden Transistoren erfolgt wechselweise. Ist der obere Transistor angesteuert, so ist der untere gesperrt und umgekehrt. Für den Anwender bedeutet dies folgendes: Ist der obere Transistor durchgeschaltet,

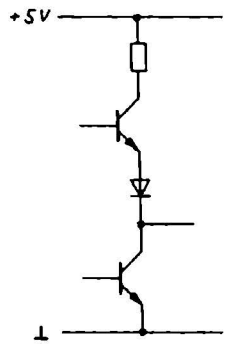


Bild 1.1.13 Aufbau einer Gegentakt-Endstufe

hat der Ausgang ein H-Signal. Ist dagegen der untere Transistor durchgeschaltet, hat der Ausgang ein L-Signal. Der obere Transistor ist so ausgelegt, daß er bei einem "fan out" von n = 10 einen Strom von ca. 1mA herausfließen lassen kann. Dieser Strom wird für die H-Signalerzeugung der angesteuerten Stufen benötigt. Bei einem H-Signal an einem TTL-Gatter fließt

ein Sperrstrom von ca. 40µA pro Gattereingang. Der Sperrstrom ist ein Reststrom, der durch die gesperrte Emitter-Basis-Diode des Eingangstransistors fließt. Dieser Strom ist stark von der Umgebungstemperatur abhängig. Der untere Transistor der Ausgangsstufe dagegen ist für einen Strom von 20mA oder 50mA je nach fan out ausgelegt. Hat der Transistor an seiner Basis ein Signal, so kann er 10 oder 30 angeschlossene Gatter gegen Masse ziehen. Der Vorteil der Gegentakt-Ausgänge ist die einfache Ansteuerung von TTL-Gattern ohne externe Widerstände.

Eintakt-Ausgänge:

Diese Ausgänge haben eine Eintakt-Endstufe und werden meistens als offener Collektor bezeichnet. Der Vorteil der Stufe ist die technische Realisierung eines Wired-AND-Gatters. Bei der Stufe liegt nur ein Transistor gegen Masse. Daher muß extern ein Arbeitswiderstand oder pull-up-Widerstand vorhanden sein. Spannungen und Ströme sind identisch wie bei der Gegentakt-Endstufe. In Bild 1.1.14 ist die Innenschaltung für diesen Ausgang gezeigt.

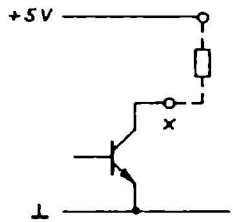


Bild 1.1.14 Aufbau einer "offenen Collektor"-Endstufe

Tri-State-Ausgänge:

Diese Art der Ausgänge ist neu. Sie bietet den Vorteil für die technische Realisierung eines Daten-BUS-Systems. In Bild 1.1.15 ist die Innenschaltung für einen Tri-State-Ausgang gezeigt. Der Ausgang bietet drei (three) Ausgangszustände:

a) Der Ausgang x hat ein H-Signal, wenn der obere Transistor leitend und der untere gesperrt ist.
b) Der Ausgang x hat ein L-Signal, wenn der untere Transistor leitend und der obere gesperrt ist.
c) Der Ausgang x hat kein definiertes Signal, wenn der obere und der untere Transistor gesperrt ist. Wir sprechen hier vom hochohmigen Zustand der Ausgangsstufe. Das Ausgangssignal floatet, d.h. es paßt sich dem Signalzustand des BUS-Systems an.

Durch den Tri-State-Ausgang können BUS-Systeme realisiert werden. Für das Bild 1.1.15 bedeutet das, hat der CS-Eingang ein H-Signal, sind die vier Ausgänge des RAMs hochohmig.

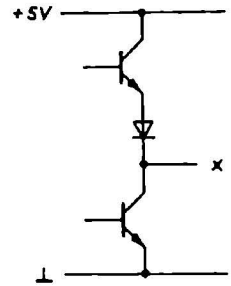


Bild 1.1.15 Aufbau einer "Tri-State"-Endstufe

Das Bild 1.1.16 zeigt ein 256-Bit-RAM mit einer Organisation von 256 x 1-Bit. Die Adressierung des Speicherplatzes erfolgt durch die A-Eingänge (Adresse). Nach der abgeschlossenen Adressierung kann eingeschrieben oder ausgelesen werden. Hierzu sind folgende Funktionen zu beachten:

Schreiben:

Die drei CE-Eingänge (chip enable) liegen auf L-Signal. Damit ist die UND-Bedingung erfüllt. An dem DI-Eingang kann eine L- oder H-Information anliegen. Durch den direkten und negierten Ausgang des DI-UND-Gatters werden die Signale für die beiden Schreib-Lese-Leitungen der Speichermatrix erzeugt. Die Einschreibung der Information erfolgt, wenn der WE-Eingang ein L-Signal hat. Liegt das Eingangssignal von dem WE-Eingang auf H-Signal, ist die Information eingeschrieben.

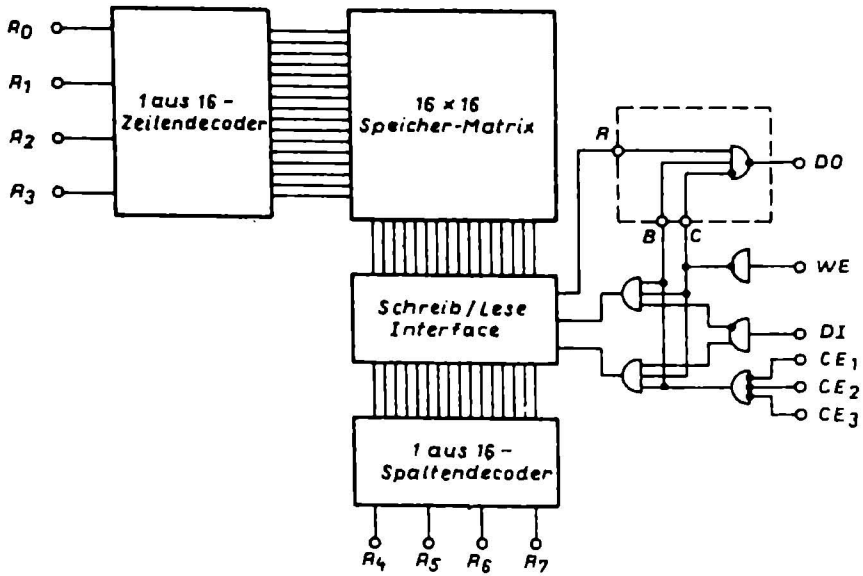


Bild 1.1.16 Aufbau eines 256-Bit-RAMs mit der Organisation von 256 x 1

Lesen:

Die drei CE-Eingänge müssen wieder auf L-Signal liegen. Der WE-Eingang erhält beim Lesen ein H-Signal. Damit kann die adressierte Speicherzelle zerstörungsfrei ausgelesen werden.

Die CE-Eingänge sind für Steuerungszwecke erforderlich. Damit kann durch eine externe Beschaltung der Schreib- und Lese-Vorgang des RAMs gesteuert werden.

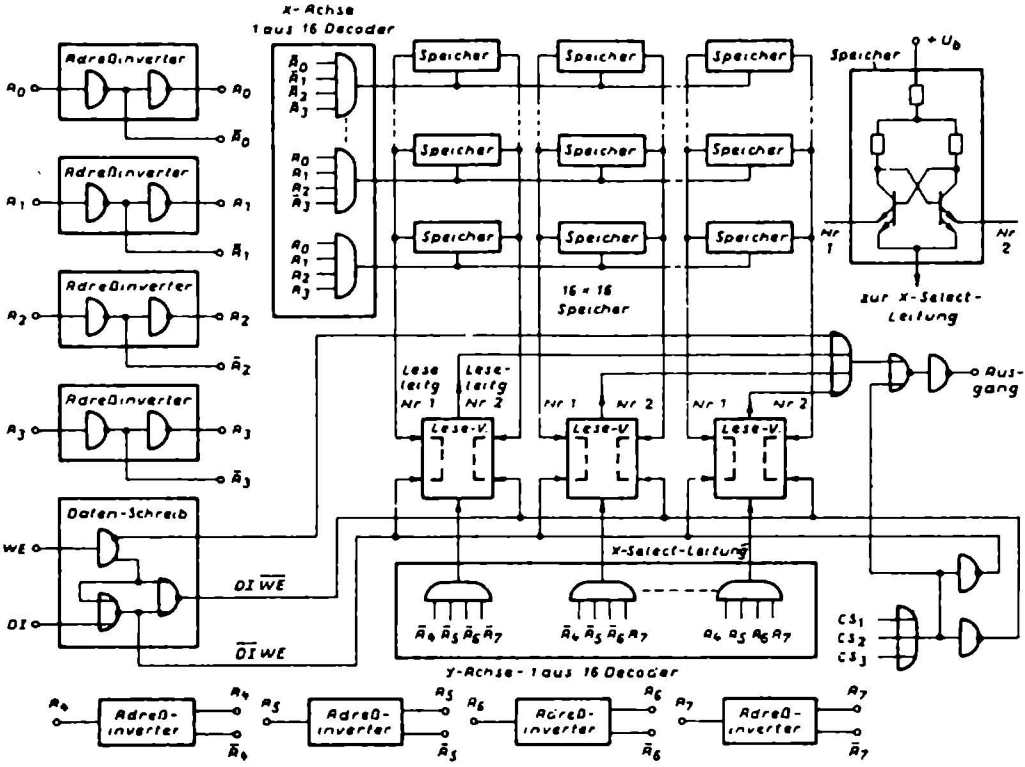


Bild 1.1.17 Interner Aufbau eines 256-Bit-RAMs

Das Bild 1.1.17 zeigt den Aufbau eines 256-Bit-RAMs. Im wesentlichen ist die Funktion mit dem Schreib-Lese-Speicher von Bild 1.1.16 identisch. Hier wird in Bild 1.1.17 die Möglickeiten der hochintegrierten TTL-Technik gezeigt. Auf einer Chip-Oberfläche von 2mm x 2mm befindet sich die gesamte Schaltung des Schreib-Lese-Speichers. An den Adressen-Eingängen A_0 bis A_7 liegen die Adressen an. Für die Adressierung wird daher ein 8-Bit-Wort, also ein Byte, benötigt. Die vier wertniedrigen Adressen A_0 bis A_3 steuert die Zeilen des RAMs an. Die vier werthöheren Adressen A_4 bis A_7 steuern die Spalten des RAMs an. Damit handelt es sich um ein volldecodiertes RAM, da jeder Speicherplatz adressiert werden kann.

Die Adressen liegen über Adreß-Invertern an den beiden Decodern an. Durch diese Inverter erreicht man ein "fan out" von n=1 für die TTL-Technik. Ebenfalls werden nach dem ersten Inverter die Negation und nach dem zweiten Inverter die Codierungssignale für die Decoder-Schaltung erzeugt. Der X-Decoder besteht aus 16 UND-Gattern mit je vier Eingängen. Das gleiche gilt auch für den Y-Decoder. Die Ausgänge der UND-Gatter des X-Decoders steuern die Zeilen der einzelnen RAM-Ebenen an. Die Ausgänge der UND-Gatter des Y-Decoders steuern die Spalten der einzelnen RAM-Senkrechten an.
Die RAM-Senkrechten enthalten 16 Spalten. An jeder Spalte befindet sich ein Schreib-Lese-Verstärker. Die beiden Funktionen arbeiten um 180° verschoben, d.h. arbeiten die 16 Schreib-Verstärker, so sind die 16 Lese-Verstärker gesperrt oder umgekehrt. Mittels den Verstärkern können die Informationen eingeschrieben oder ausgelesen werden.

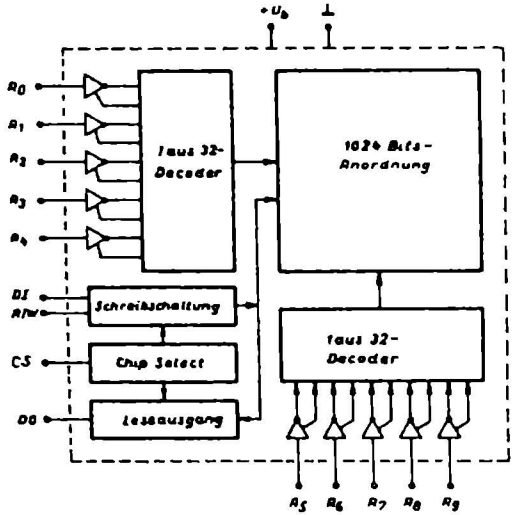


Bild 1.1.18 Aufbau eines 1-k-RAMs

Bei größeren Speicherkapazitäten verwendet man 1-k-RAMs. In Bild 1.1.18 ist ein RAM mit 1024 Speicherplätzen gezeigt. Die Adressierung erfolgt hier über 10 Adresseneingänge A_0 bis A_9. Für die Adressierung wird daher ein 10-Bit-Wort benötigt.
In der Mikroprozessorentechnik findet man häufig Schreib-Lese-Speicher mit 1-k-Struktur. Durch diese Struktur wird ein 2-Adressen-Byte benötigt. Beim Mikroprozessor ergibt das einen 3-Byte-Befehl:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Operations-Byte
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	niederwertige Adresse
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	höherwertige Adresse

In dem Operations-Byte (Op-Code) befindet sich der betreffende Befehl. Typische Befehle für diesen 3-Byte-Befehl sind Sprung-, Aufruf für ein Unterprogramm oder direkter Lade- oder Speicher-Befehl.
Mit dem ersten Byte erhält der Mikroprozessor seinen Befehl und mit den beiden anderen Bytes seine Adresse. Mit dem zweiten Byte (niederwertige Adresse) kann er das RAM von den Eingängen A_0 bis A_7 adressieren. Die beiden anderen Adressen-Eingänge werden durch das höherwertige Adressen-Byte adressiert.
Das Einschreiben oder Auslesen von Informationen wird wie bei den anderen TTL -RAMs durchgeführt. Mit einem L-Signal an dem CS-Eingang wird der Baustein blockiert. Danach kann die Adressierung erfolgen. Ist die Adressierung abgeschlossen, müssen die Daten zum Einschreiben anliegen. Mit einem H-Signal an dem CS-Eingang und R/W-Eingang erfolgt das Einschreiben der Information, die an dem DI-Eingang anliegt. Das Auslesen kann nur erfolgen, wenn die Adressierung abgeschlossen ist und der CS-Eingang auf H-Signal und der R/W-Eingang auf L-Signal liegt. Die gespeicherte Information liegt an dem DO-Ausgang.
Reicht die Speicherkapazität eines 1-k-RAMs nicht mehr aus, kann die Speicherkapazität einfach erweitert werden. In Bild 1.1.19 ist eine Möglichkeit gezeigt. Hier wurden vier 1-k-RAMs zusammengeschaltet. Für die Adressierung wird hier noch ein Decoder benötigt. An dem 1- aus 4-Decoder liegen die Adressen A_{10} und A_{11} an. Es ergibt sich daher für den Betrieb folgende Funktionstabelle:

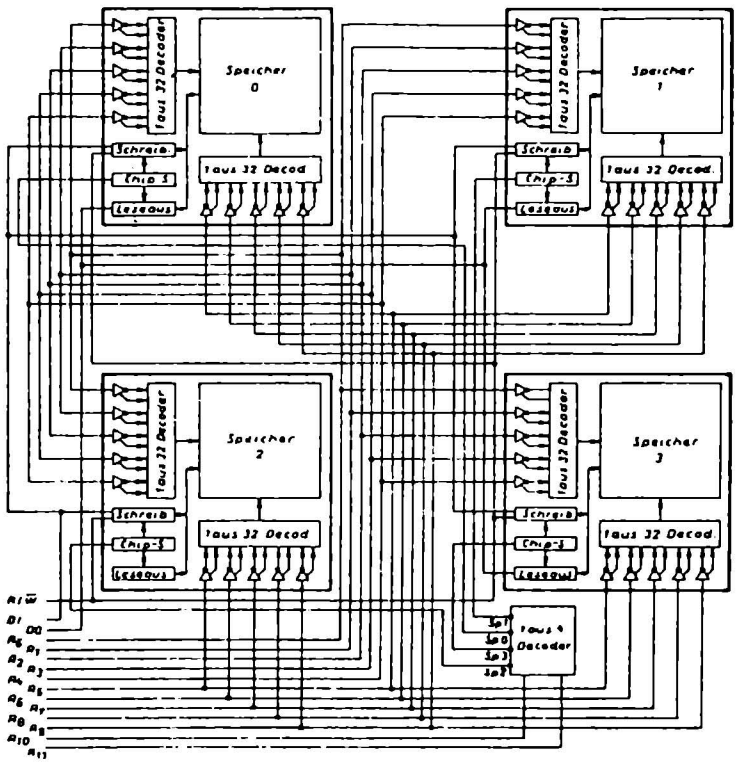


Bild 1.1.19 Aufbau eines 4-k-RAMs

Speicherplätze	Adressen	A_{11}	A_{10}	Speicher
0 bis 1023	0 bis 1023	L	L	0
1024 bis 2047	1024 bis 2047	L	H	1
2048 bis 3071	2048 bis 3071	H	L	2
3072 bis 4095	3072 bis 4095	H	H	3

Die Adressierung dieser Speicherkapazität durch den Mikroprozessor stellt kein Problem dar.

Durch den Decoder kann der Mikroprozessor jeden Speicherplatz innerhalb des 4-k-Speichers erreichen. Für die Adressierung werden die D_2 und D_3-Stelle des höherwertigen Adressen-Bytes benötigt.

Seit kurzer Zeit sind 4-k-RAMs auf einem Chip erhältlich. Hierbei reduziert sich der Bauelementeaufwand von 5 Bausteinen (vier 1-k-RAMs und ein Decoder) auf nur einen Baustein. Der Baustein ist mit dem Aufbau des 1-k-RAMs von Bild 1.1.18 weitgehend identisch. Die X-Achsen werden durch sechs Adressen-Eingänge A_0 bis A_5 und die Y-Achsen durch die sechs Adressen-Eingänge A_6 bis A_{11} angesteuert.

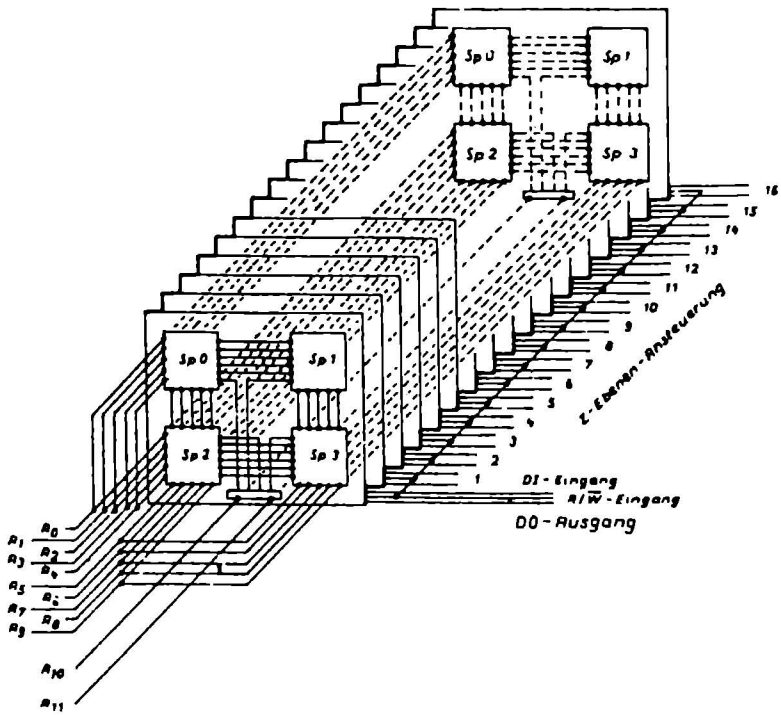


Bild 1.1.20 Großspeicher für 65 536 Worte x 1-Bit

Das Bild 1.1.20 zeigt einen Großspeicher für einen Mikroprozessor. Die Schaltung besteht aus 16 Schaltungen von Bild 1.1.19. Die Adressierung der Adressen A_0 bis A_{11} erfolgt parallel über alle 16 Ebenen. Die einzelnen Ebenen werden durch die Z-Ansteuerung vorgenommen. Die Adressen A_{12} bis A_{15} steuern einen 4- aus 16-Decoder an. Hierzu werden die D_4 bis D_7-Stellen des höherwertigen Adressen-Bytes benötigt. Der Decoder decodiert die vier Stellen in 16 einzelne Leitungen für jede Ebene.
In der Schaltung von Bild 1.1.20 können 65 536 Wörter x 1 Bit gespeichert werden. Das Einschreiben einer Information ist von dem R/W-Eingang abhängig. Eine CS-Steuerung ist nicht möglich, da diese Eingänge für die Ansteuerung durch die 1- aus 4-Decoder bereits benötigt werden. Hat der R/W-Eingang ein L-Signal, wird eingeschrieben und bei einem H-Signal ausgelesen.

1.2. Statische Schreib-Lese-Speicher in MOS-Technik

Die MOS-Technik bietet gegenüber der TTL-Technik zahlreiche Vorteile, wie geringe Leistungsaufnahme und hohe Packungsdichte. Bei der MOS-Technik unterscheiden wir zwischen der p- und n-Kanal-Technik.

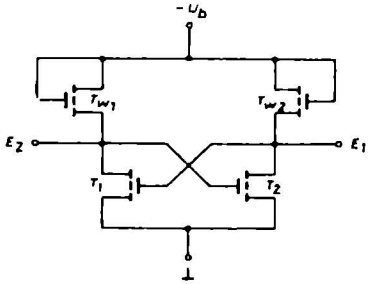


Bild 1.2.1 Prinzip einer statischen MOS-Speicherzelle in p-Kanal-Technologie

p-Kanal-Technik:

In Bild 1.2.1 ist das Prinzip einer statischen MOS-Speicherzelle in p-Kanal-Technik gezeigt. Die beiden oberen MOS-Transistoren sind als Arbeitswiderstände geschaltet. Über die beiden Eingänge E_1 und E_2 wird die p-Kanal-Speicherzelle angesteuert. Geben wir auf den E_1-Eingang ein L-Signal ($-U_b$), so wird der Transistor T_1 leitend, d.h. er baut zwischen Drain und Source einen Kanal auf. Durch den Kanal wird der Eingang E_2 mit Masse verbunden und hat daher ein H-Signal. Das H-Signal liegt durch die kreuzweise Verbindung der beiden Gate-Anschlüsse an dem Gate des Transistors T_2 an und sperrt diesen. Nehmen wir von dem Eingang E_1 das L-Signal, bleibt die Speicherzelle in diesem Zustand. Die Speicherzelle kann mit einem L-Signal ($-U_b$) an dem Eingang E_2 gekippt werden. Dadurch wird der Transistor T_2 leitend, d.h. er baut zwischen Drain und Source einen Kanal auf. Durch den Kanal wird der Eingang E_1 mit Masse verbunden und hat daher ein H-Signal. Das H-Signal liegt durch die kreuzweise Verbindung der beiden Gate-Anschlüsse an dem Gate des Transistors T_1 an und sperrt diesen. Nehmen wir von dem Eingang E_1 das L-Signal, so bleibt dieser Zustand gespeichert.

n-Kanal-Technik:

In Bild 1.2.2 ist das Prinzip einer statischen MOS-Speicherzelle in n-Kanal-Technik gezeigt. Die Betriebsspannung beträgt nicht $-U_b$, sondern $+U_b$. Die beiden oberen MOS-Transistoren sind als Arbeitswiderstände geschaltet. Legen wir an einem n-Kanal-MOS-Transistor ein L-Signal an, so wird der Kanal zwischen Drain und Source-Anschluß abgebaut. Dagegen kann sich zwischen Source und Drain ein Kanal aufbauen, wenn ein H-Signal angelegt wird.
Legen wir an den Eingang E_1 ein H-Signal an, so wird der Transistor T_1 angesteuert, d.h. er baut einen Kanal auf und verbindet damit den Eingang E_2 mit Masse (L-Signal). Durch die kreuzweise Verbindung der beiden Gate-Anschlüsse wird der Transistor T_2 gesperrt. Nehmen wir das H-Signal von dem Eingang E_1, so bleibt die Information gespeichert. Wir können die Information ändern, wenn wir an den Eingang E_2 ein H-Signal anlegen. Dadurch baut sich in dem Transistor T_2 ein Kanal auf, der den Eingang E_1 mit Masse (L-Signal) verbindet. Durch das L-Signal wird der Transistor T_1 gesperrt. Nehmen wir das H-Signal von dem Eingang E_2, so bleibt die Information gespeichert.

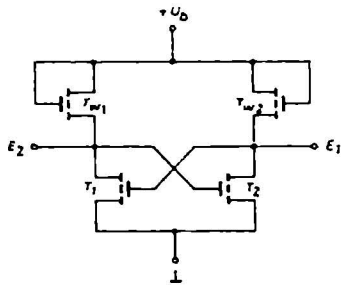


Bild 1.2.2 Prinzip einer statischen MOS-Speicherzelle in n-Kanal-Technologie

In der heutigen Fertigung werden fast nur noch n-Kanal-MOS-RAMs hergestellt. In Bild 1.2.3 ist eine Sechs-Transistor-Speicherzelle gezeigt. Durch die x-Adreß-Leitung ergibt sich ein Zeilen-orientiertes-RAM.

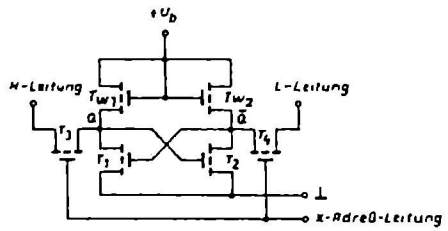


Bild 1.2.3 Aufbau einer sechs-Transistoren-Speicherzelle in n-Kanal-Technik

Für ein Zeilen-orientiertes-RAM sind sechs MOS-Transistoren für die technische Realisierung einer MOS-Speicherzelle erforderlich. Die beiden Transistoren T_3 und T_4 dienen zur Adressierung der Speicherzelle. Bei einem H-Signal kann sich in beiden Transistoren ein Kanal aufbauen und damit eine elektrische Verbindung von der H- oder L-Leitung zu der eigentlichen Speicherzelle herstellen lassen. Wir erhalten mit einem H-Signal auf der x-Adreß-Leitung damit eine adressierte Speicherzelle.
Das Einschreiben einer Information geht folgendermaßen vor sich. Der Q-Punkt in dem Bild 1.2.3 soll ein H-Signal und der $\bar{Q}$-Punkt ein L-Signal aufweisen. Daher ist der Transistor T_1 gesperrt und der Transistor T_2 leitend. Die H- und L-Leitung müssen sich hierbei vor der Adressierung ändern. In diesem Fall muß die H-Leitung ein L-Signal haben. Bei der Adressierung liegt das H-Signal des Q-Punktes auf dem L-Signal der H-Leitung über den Transistor T_3. Der Punkt Q wird gegen Masse geschaltet und daher erhält der Transistor T_2 ein L-Signal und sperrt, d.h. der Kanal wird abgebaut. Der Punkt $\bar{Q}$ geht auf H-Signal und der Transistor T_1 beginnt zu leiten, d.h. der Transistor T_1 baut einen Kanal auf. Geht die Adressierung wieder auf L-Signal, sind die beiden Transistoren T_3 und T_4 wieder gesperrt.
In der Schaltung von Bild 1.2.3 ist die Funktionsweise der Transistoren T_3 und T_4 interessant. Je nach Signal auf den Leitungen verändern sich Source und Drain. Dies ist von der Information auf den Leitungen abhängig. Durch die entsprechenden Informationen werden Source und Drain je nach Polarität vertauscht. Hier wird die vollkommene Symmetrie der MOS-Transistoren ausgenutzt, wonach sich Source und Drain einzig nach der Polarität der anliegenden Spannung einstellen.

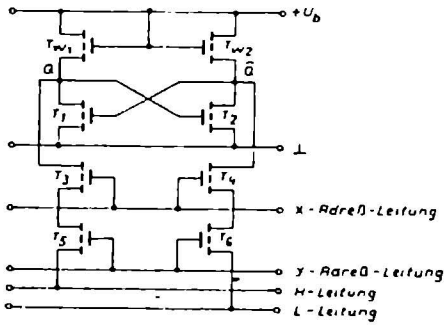


Bild 1.2.4 Aufbau einer acht-Transistoren-Speicherzelle in n-Kanal-Technik

Die Speicherzelle von Bild 1.2.3 läßt sich einfach erweitern. In Bild 1.2.4 wurden noch zwei MOS-Transistoren hinzugefügt. Dadurch erhalten wir ein Bit-orientiertes-RAM. Zur besseren Erklärung verwenden wir n-Kanal-Transistoren. Die Adressierung der Speicherzelle kann nur erfolgen, wenn an der x- und y-Adreß-Leitung ein H-Signal anliegt. Von den vier Adressierungstransistoren bilden immer zwei ein NAND-Gatter, T_3 und T_5, bzw. T_4 und T_6. Haben beide Leitungen ein H-Signal, so haben alle vier Transistoren einen Kanal aufgebaut und die Punkte Q

und $\bar{Q}$ der Speicherzelle liegen auf der H- und L-Leitung. Die NAND-Gatter-Funktion ist durch die Phasendrehung von 180° der in Reihe geschaltenen MOS-Transistoren bedingt.
In Bild 1.2.5 ist ein voll decodiertes statisches RAM in MOS-Technik gezeigt. Über die acht Eingänge erfolgt die Adressierung der 256 Speicherplätze. Dadurch ergibt sich eine Organisation von 256 x 1 Bit, d.h. wir haben 256 Speicherplätze für ein Bit. Die Adressen liegen über Buffer an den beiden X- und Y-Decodern an. Die Daten werden über den DI-Eingang eingeschrieben. Mit einem L-Signal an dem R/W-Eingang kann die Information auf den adressierten Speicherplatz eingeschrieben werden. Das Ausschreiben einer gespeicherten Information ist nur möglich, wenn der R/W-Eingang an H-Signal liegt. Dabei kann an dem DO-Ausgang die identische

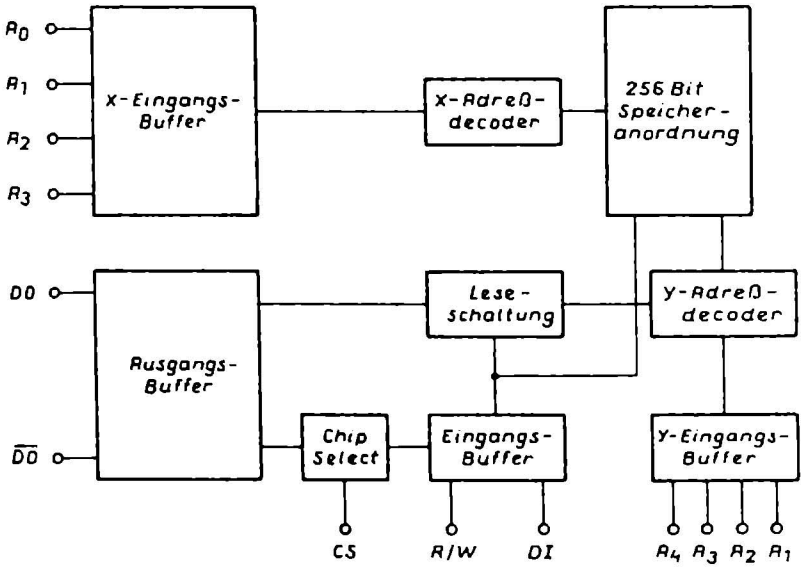


Bild 1.2.5 Statischer Schreib-Lese-Speicher in n-Kanal-MOS-Technik

Eingangsinformation und an dem $\overline{DO}$-Ausgang die invertierte Eingangsinformation abgenommen werden. Wichtig beim Einschreiben und Auslesen ist der Zustand des CS-Einganges. Bei einem L-Signal kann weder eingeschrieben, noch ausgelesen werden.
In Bild 1.2.6 ist die Innenschaltung für dieses RAM gezeigt. An den X-Eingangs-Buffern liegen die Adreß-Eingänge an. Mit diesen vier Eingängen werden die Zeilen der Speicherplätze selektiert. Die Spalten der Speicherplätze werden durch die Adressen-Eingänge angesteuert. Damit erhalten wir ein volldecodiertes RAM.
Die Eingangsinformation liegt an dem DE-Eingang (Daten-Eingang oder DI) an. Da nur ein Eingang vorhanden ist, kann in das RAM immer nur ein Bit eingeschrieben werden. Für das Einschreiben einer Information muß der CS-Eingang auf L-Si-

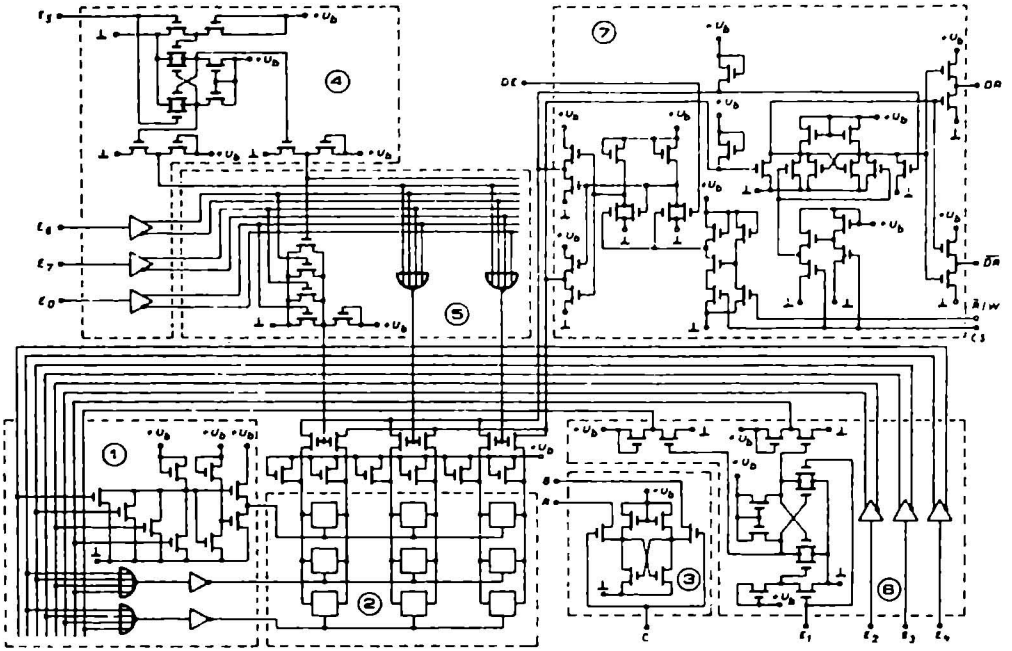


Bild 1.2.6 Innenschaltung eines 256-Bit-Schreib-Lese-Speichers in p-Kanal Technik

gnal gelegt werden, ebenso der R/W-Eingang. Für die einzelnen Funktionsblöcke in dem Bild 1.2.6 ergeben sich folgende Teile:

Block 1:

Dieser Teil zeigt den X-Adressen-Decoder. Die Decodierung wird mittels eines NOR-Gatters und einem nachgeschalteten NICHT-Gatter erzeugt. Dies ist durch die p-Kanal-Technik bedingt, aber für den Anwender ist dies nicht interessant. Jedes NOR-Gatter besteht aus vier MOS-Transistoren, die parallel geschaltet sind. Die beiden anderen Transistoren invertieren das Ausgangssignal. Damit ergibt sich ein ODER-Verhalten. Da es sich aber um p-Kanal-MOS-Transistoren handelt, tritt hier eine UND-Funktion auf.

Block 2:

Dieser Teil ist die Speicheranordnung und besteht aus 256 Speicherzellen. Die Anordnung der Speicherplätze wurde mit 16 in X- und mit 16 in Y-Richtung festgelegt. Dadurch ergeben sich 16x16=256 Speicherplätze. Die Adressierung der X-Reihe oder Zeilen erfolgt durch den X-Decoder und die Y-Reihe oder Spalten durch den Y-Decoder. Daher ergibt sich ein volldecodiertes RAM.

Block 3:

In diesem Funktionsblock ist der Aufbau einer statischen MOS-Speicherzelle nochmals gezeigt.

Block 4:

Der Y-Adressen-Buffer besteht aus vier Gattern mit einem invertierenden und einem nichtinvertierenden Ausgang. Die Ausgänge sind direkt mit dem Block 5, dem Y-Adressen-Decoder verbunden.

Block 5:

Mit dem Y-Adressen-Decoder werden die Spalten der Speicheranordnung angesteuert. Der Decoder besteht nur aus NOR-Gattern, was durch die p-Kanal-Technik bedingt ist. Jedes NOR-Gatter besteht aus vier MOS-Transistoren und einem MOS-Transistorwiderstand.

Block 6:

Der X-Adressen-Eingang-Buffer liegt zwischen den Eingängen und dem Decoder und erzeugt einen invertierenden und einen nichtinvertierenden Ausgang.

Block 7:

In diesem Funktionsblock befindet sich die gesamte Ansteuerungslogik für das MOS-RAM, mit dem Daten-Eingang DE (oder DI), dem direkten Daten-Ausgang DA (oder DO), dem invertierenden Daten-Ausgang $\overline{DA}$ (oder $\overline{DO}$) und dem CS-Eingang (chip select).

Das Bild 1.2.7 zeigt das Blockschaltbild eines 1024-Bit-MOS-RAMs mit der Organisation 256 x 4. Hierbei handelt es sich um den Typ 2101. Der Baustein ist direkt

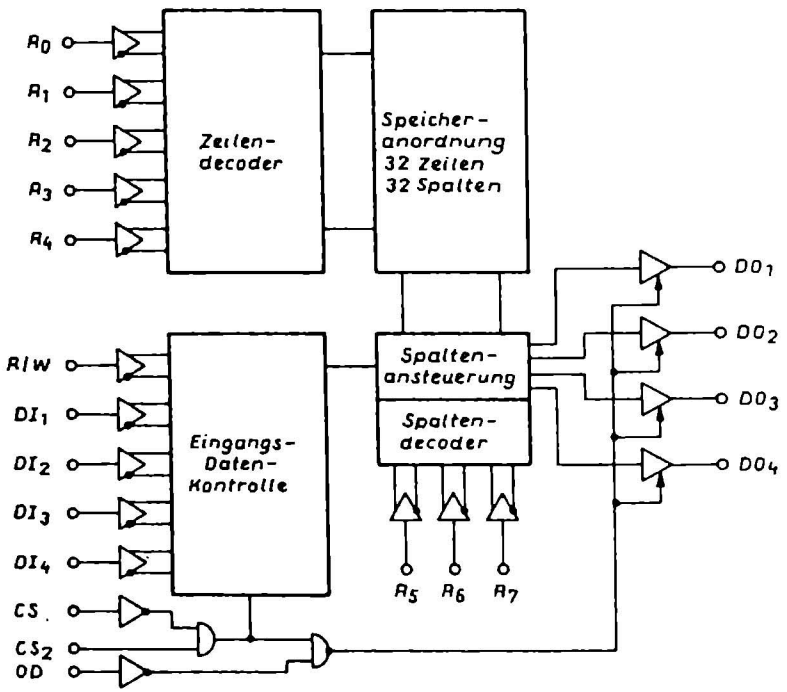


Bild 1.2.7 MOS-RAM 2101, 1024-Bit mit der Organisation von 256x4

TTL-kompatibel in allen Anschlüssen, wie Eingänge, Ausgänge und Betriebsspannung. Durch die beiden CS-Eingänge können wir externe Steuerungen aufbauen oder diese Eingänge für Erweiterungszwecke verwenden.
Das RAM in Bild 1.2.7 weist 32 Zeilen und 32 Spalten auf. Die 32 Zeilen werden von den fünf Adressen-Eingängen angesteuert. Zwischen den Eingangs-Buffern und der Speichermatrix befindet sich ein 1- aus 32-Decoder. Hiermit kann jede der 32 Zeilen angesteuert werden. Die 32 Spalten werden durch die drei höherwertigen Adressen selektiert. Die drei Eingänge erzeugen durch einen 1- aus 8-Decoder insgesamt acht Spalten. Durch die Struktur der vier Eingangs-Bits ergibt sich pro Spalte ein vierstelliges Wort und damit erhalten wir 8 x 4 = 32 Spalten für die Speichermatrix.

Die Bezeichnungen in Bild 1.2.7 lauten aufgeschlüsselt:

DI = Daten-Eingänge (data input)

DO = Daten-Ausgänge (data output)

R/W = Schreib-Lese-Steuerung (read-write-input)

CS = Steuerung (chip select)

OD = Ausgangssteuerung (output disable)

A = Adressen-Eingänge (address inputs)

Bei einem L-Signal an dem R/W-Eingang können wir ein 4-Bit-Datenwort einschreiben und bei einem H-Signal auslesen.
Der CS_1-Eingang ist auf L-Signal und der CS_2-Eingang auf H-Signal zu legen, damit die UND-Bedingung erfüllt ist. Bei diesen Eingangsbedingungen kann in das RAM eingeschrieben oder der Inhalt ausgelesen werden.
Mit der Ausgangssteuerung können wir die Ausgangs-Buffer steuern. Bei Auslesen einer Information muß bereits die UND-Bedingung erfüllt sein. Damit liegt an dem NAND-Gatter ein H-Signal an. Hat der OD-Eingang ein L-Signal, so wird dieses durch das NICHT-Gatter invertiert und liegt mit einem H-Signal an dem NAND-Gatter an. Daher ist die NAND-Gatter-Bedingung erfüllt und die Ausgangsinformationen können auf die Ausgänge geschaltet werden. Durch den OD-Eingang kann das RAM an ein BUS-System angeschlossen werden, da durch diesen Eingang die Ausgänge steuerbar sind.
Die Ausgänge des RAM-Bausteines 2101 haben einen Tri-State-Ausgang, d.h. hat der OD-Eingang ein H-Signal, so sind die-Ausgänge des Bausteines hochohmig. Nur wenn die drei Eingangsbedingungen CS_1, CS_2 und OD erfüllt sind, können gespeicherte Informationen auf den Daten-BUS gegeben werden.
Das RAM von Bild 1.2.7 ist ein Wort-orientierter-Schreib-Lese-Speicher, dagegen ist das RAM von Bild 1.2.8 ein Bit-orientierter-Schreib-Lese-Speicher. Es handelt sich um den Baustein 2102. Die Adressierung erfolgt durch die zehn Adressen-Eingänge A_0 bis A_9. Die fünf Adressen-Eingänge A_0 bis A_4 steuern die Zeilen und die anderen fünf die Spalten an. Dadurch ergibt sich ein voll-decodiertes RAM mit einer Organisation von 1024 x 1-Bit.
Die Steuerung des Tri-State-Ausganges erfolgt über den CS-Eingang. Hat der Eingang ein H-Signal, ist der Ausgang hochohmig. Gleichzeitig kann das gesamte RAM durch das L-Signal blockiert werden.

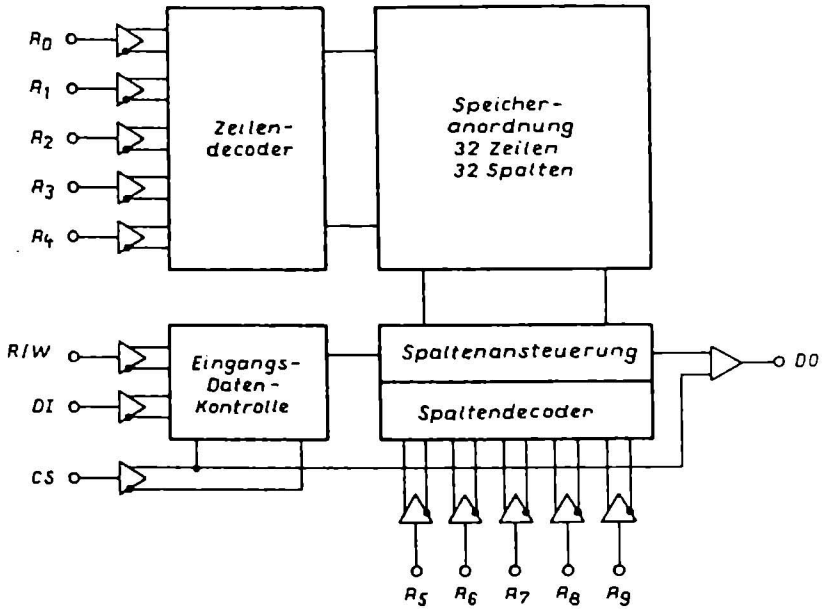


Bild 1.2.8 MOS-RAM 2102, 1024-Bits mit der Organisation von 1024x1

Mit einem L-Signal an einem R/W-Eingang erfolgt das Einschreiben von einem Daten-Bit auf den DI-Eingang. Die eingeschriebenen Daten-Bits können ausgelesen werden, wenn an dem R/W-Eingang ein H-Signal anliegt. Durch den Tri-State-Ausgang können diese RAMs an einen Daten-BUS angeschlossen werden.
Das Bild 1.2.9 zeigt einen Schreib-Lese-Speicher mit gemeinsamen Daten Ein- und Ausgängen. An den I/O-Punkten (input/output) können die anstehenden Daten in das RAM eingeschrieben oder die gespeicherten Daten aus dem RAM ausgelesen werden. Der Einschreibe- und Auslesebetrieb ist beim Schreib-Lese-Speicher vom Typ 2111 ein Problem der Ansteuerung.
Die Ansteuerung des Bausteins 2111 wird durch die drei Eingänge CS_1, CS_2 und OD vorgenommen. Der Einschreibevorgang läuft folgendermaßen ab. Einer der beiden CS-Eingänge muß auf H-Signal liegen, ebenso der OD-Eingang für die Ausgangssteuerung. Ist die Adressierung abgeschlossen, erhält der betreffende CS-Eingang ein L-Signal. Damit meldet ein Steuerbaustein dem Mikroprozessor, daß die Adressierung abgeschlossen ist. Der Mikroprozessor gibt seinerseits einen Freigabe-Befehl an das RAM ab und zwar auf einen der beiden CS-Eingänge.
Nachdem beide CS-Eingänge auf L-Signal liegen, ist die UND-Bedingung erfüllt und das anstehende 4-Bit-Wort wird auf den adressierten Speicherplatz eingeschrieben. Durch das H-Signal an dem OD-Eingang ist die NAND-Gatter-Bedingung nicht erfüllt, da sich vor dem NAND-Gatter ein NICHT befindet. Das NAND-Gatter hat an seinem Ausgang ein H-Signal und steuert damit die Ausgangs-Buffer an. Ein H-Signal aber bedeutet, daß die Ausgangsstufe des Tri-State-Ausganges hochohmig ist.
Werden bei dem Baustein 2111 die Daten ausgelesen, so müssen die CS-Eingänge

auf L-Signal liegen. Der auslösende Impuls kann der OD-Eingang sein. Mit einem L-Signal ist die NAND-Gatter-Bedingung erfüllt und der NAND-Gatter-Ausgang

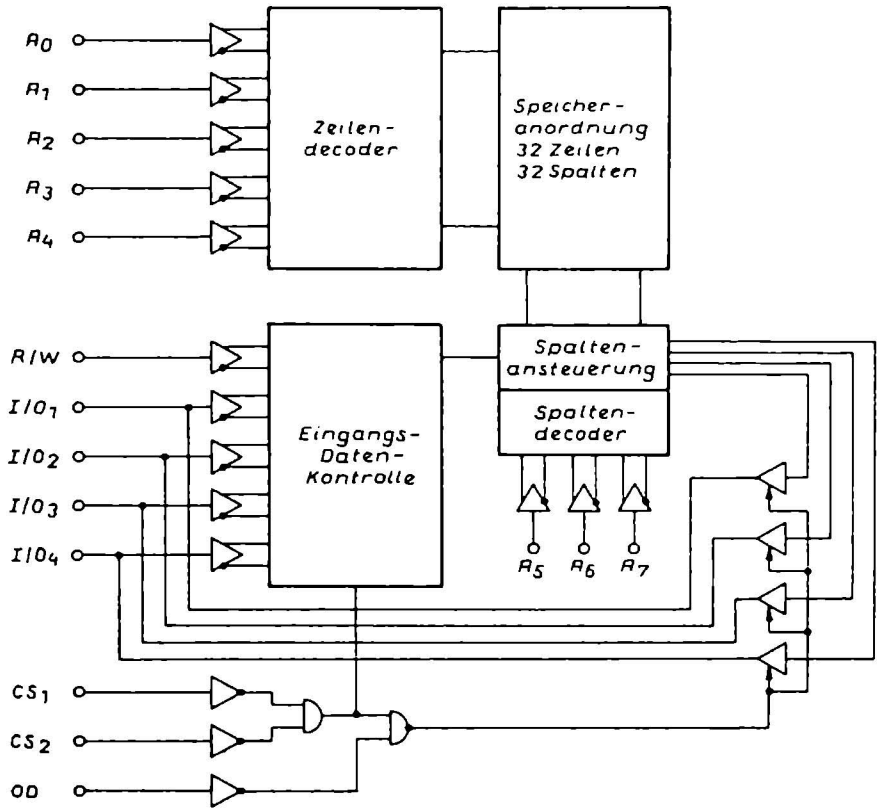


Bild 1.2.9 MOS-RAM 2111, 1024-Bits mit der Organisation von 256 x 4

hat ein L-Signal. Für den Ausgangs-Buffer bedeutet dies, daß die adressierten vier Speicherzellen an dem Ausgang anliegen.
Zweckmäßigerweise steuert der Mikroprozessor einen der CS-Eingänge an. Der OD-Eingang kann durch eine Codierung ebenfalls von dem Mikroprozessor angesteuert werden.
Das RAM von Bild 1.2.10 ist ein 1024-Bit-RAM mit einer Organisation von 256 x 4 Bit. Wesentlich ist hierbei die Speicherung der Adresse in den beiden Adressen-Registern, durch den LATCH-Eingang. Liegt eine Adresse an, so kann diese durch ein L-Signal gespeichert werden. Damit liegt die Adresse an und bleibt solange erhalten, bis der LATCH-Eingang auf H-Signal geht. Mit einem H-Signal werden die Adressen direkt auf die Speicheranordnung durchgeschaltet.
In das RAM von Bild 1.2.10 kann eingeschrieben werden, wenn die Adresse an den Decodern anliegt und der CS-Eingang ein L-Signal hat. Der WE-Eingang muß zu diesem Zeitpunkt ein L-Signal haben.

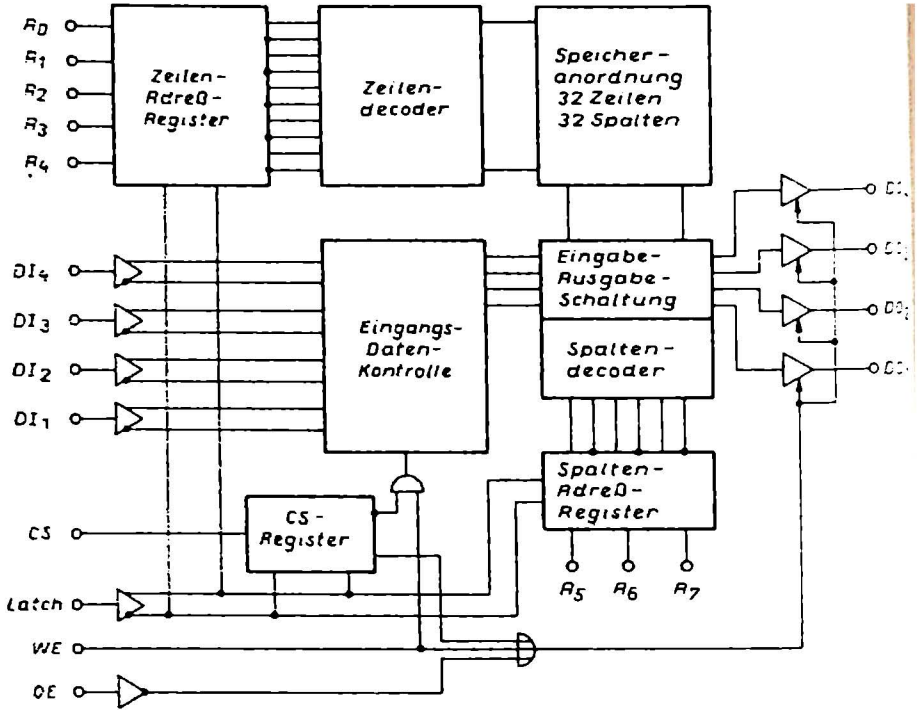


Bild 1.2.10 MOS-RAM 5269, 1024-Bits mit der Organisation von 256x4

Das Auslesen der Daten kann durch ein H-Signal an dem WE-Eingang erfolgen oder an dem OE-Eingang liegt ein L-Signal an. Die Ausgangs-Buffer schalten die gespeicherte Information auf den Ausgang. Die Ausgangssteuerung für den hochohmigen Zustand ist nur dann vorhanden, wenn alle Eingänge des ODER-Gatters ein L-Signal aufweisen. Dadurch ist die ODER-Bedingung nicht erfüllt und der ODER-Ausgang hat ein L-Signal. Die Ausgänge des Schreib-Lese-Speichers sind damit hochohmig.

1.3. Dynamische Schreib-Lese-Speicher in MOS-Technik

Dynamische Schreib-Lese-Speicher in MOS-Technik speichern ihre Information in Form einer Ladung in einem Kondensator, d.h. ist der Kondensator geladen, so haben wir ein H-Signal und ist er entladen, ein L-Signal.
In Bild 1.3.1 ist eine dynamische MOS-Speicherzelle gezeigt. Wir haben hier vier Transistoren und zwei Kondensatoren. Die Transistoren T_1 und T_4 dienen zur Adressierung der Speicherzelle. Die beiden anderen Transistoren sind die eigentlichen Speicher-Transistoren und dienen für den Betrieb. In den beiden Kondensatoren ist der Informationszustand der Zelle gespeichert.
Wir wollen ein Signal einschreiben. Die H-Leitung erhält in Bild 1.3.1 ein H-Signal, die L-Leitung ein L-Signal. Hat die x-Linie ein H-Signal, so ist die Speicher-

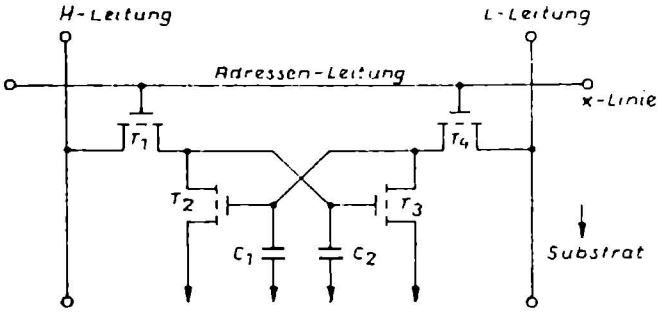


Bild 1.3.1 Aufbau einer dynamischen MOS-Speicherzelle mit vier Transistoren und zwei Kondensatoren

zelle adressiert. Über den Transistor T_1 kann sich der Kondensator C_2 aufladen. Nach ca. 0,7 τ erreicht die Kondensatorladung den halben Wert der Betriebsspannung und der Transistor T_3 schaltet durch. Damit ist das Gate von dem Transistor T_2 über den Kanal von T_3 mit dem Substrat, der Masse verbunden. Der Transistor T_2 ist gesperrt und der Kondensator C_1 weist keine Ladung auf. In die Speicherzelle wurde ein H-Signal eingeschrieben.

In die Speicherzelle von Bild 1.3.1 soll ein L-Signal eingeschrieben werden. Die L-Leitung geht auf H-Signal und die x-Linie erhält ebenfalls ein H-Signal. In dem Transistor T_1 und T_4 baut sich ein Kanal auf. Da an dem Kondensator C_2 in diesem Fall ein L-Signal durch die H-Leitung anliegt, entlädt sich der Kondensator C_2. Durch das H-Signal an der L-Leitung kann sich dagegen der Kondensator C_1 aufladen. Bei ca. 0,7 τ erreichen beide fast gleichzeitig den Schwellwert für den Umschaltpunkt. Der Transistor T_2 geht in den leitenden und der T_3 in den sperrenden Zustand über. In der Speicherzelle wurde ein L-Signal gespeichert.

Ein wesentliches Problem bei dynamischen Schreib-Lese-Speichern ist das Auffrischen der Gate-Kapazitäten. Die Kapazität der beiden Kondensatoren liegt in der Größenordnung von 0,1pF bei einer Umgebungstemperatur von 25°C. Ist der Gate-Kondensator aufgeladen, so fließt ein Leckstrom von ca. 0,1pA. Dies gilt aber auch, wenn der Kondensator entladen ist. Damit kann sich der Kondensator auf- oder entladen.

Wegen der Leckströme muß der Speicherinhalt in den Kondensatoren periodisch regeneriert (refresh) werden. Die Regenerierung der Kondensatoren erfolgt im Baustein durch die Lesezyklen auf der gesamten Zeile.

Bei einer Umgebungstemperatur von 25°C muß jeder Kondensator innerhalb von 1ms aufgefrischt werden, da sonst die Information verloren geht. Steigt die Umgebungstemperatur an, so vergrößern sich die Leckströme und die Auffrischung der Speicherzellen muß daher in kürzeren Zeitintervallen erfolgen.

Die Weiterentwicklung in der MOS-Technologie brachte die Speicherzelle von Bild 1.3.2. Für diese MOS-Speicherzelle werden nur drei MOS-Transistoren und ein Kondensator verwendet.

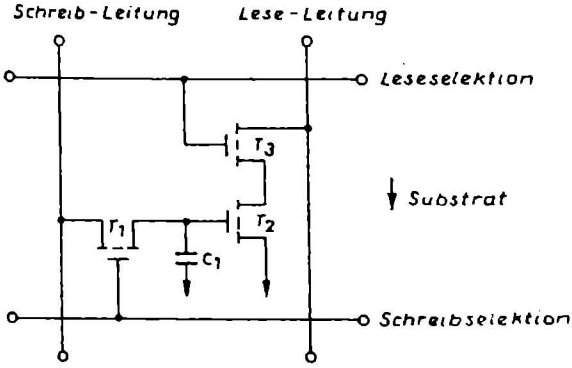


Bild 1.3.2 Aufbau einer dynamischen MOS-Speicherzelle mit drei Transistoren und einem Kondensator

In die Speicherzelle kann ein H-Signal eingeschrieben werden, wenn an der Schreibselektion und an der Schreib-Leitung je ein H-Signal anliegt. Dadurch wird in dem MOS–Transistor T_1 ein Kanal aufgebaut und der Kondensator C_1 kann sich aufladen. Hat die Schreib-Leitung ein L-Signal, kann sich der Kondensator entladen, vorausgesetzt, die Schreibselektion liegt auf H-Signal.
Für den Transistor T_2 in der Speicherzelle gilt folgende Bedingung. Ist in der Zelle ein H-Signal gespeichert, d.h. der Kondensator C_1 ist geladen, so hat der Transistor T_2 am Gate ein H-Signal und ist daher durchgeschaltet. Ist der Kondensator C_1 dagegen entladen, da er ein L-Signal gespeichert hat, ist der Transistor T_2 gesperrt. Der untere Anschluß des Transistors ist mit dem Substrat, also mit der Masse verbunden.
Beim Lesevorgang wird durch die Leseselektion die Speicherzelle mit einem H-Signal angesteuert. Dadurch ist der Transistor T_3 leitend, da sich bei einem H-Signal am Gate ein Kanal aufbauen kann. Ist der Transistor T_2 durchgeschaltet, da am Gate eine Ladung anliegt, so erkennt der Leseverstärker auf der Lese-Leitung ein L-Signal an. Das L-Signal wird durch den Leseverstärker invertiert und wir erhalten am Ausgang ein H-Signal, also die gespeicherte Information in der Speicherzelle. Hat aber die Speicherzelle ein L-Signal gespeichert, so ist der Transistor T_2 gesperrt. Die x-Lese-Leitung erkennt ein H-Signal an und dieses liegt über dem Leseverstärker invertiert an dem Ausgang an. Damit erkennen wir ein gespeichertes L-Signal in einer Speicherzelle.
Die Speicherzelle von Bild 1.3.3 besteht aus drei MOS-Transistoren und einem Kondensator C. Die Speicherzelle besteht aus einer Schreib- und Lese-Takt Leitung. Die beiden Takt-Leitungen stellen eine Selektions-Leitung dar. Beim Einschreiben oder Auslesen einer Information wird die Schreib-Lese-Leitung verwendet.
In die Speicherzelle von Bild 1.3.3 soll ein H-Signal eingeschrieben werden. Hierzu erhält die gemeinsame Schreib-Lese-Leitung ein H-Signal. Danach schalten wir auf die Schreib-Takt-Leitung ein H-Signal. Der Transistor T_3 baut einen Kanal auf und der Kondensator C lädt sich durch das H-Signal auf. In der Speicherzelle ist eine Information gespeichert, die Speicherzelle hat ein H-Signal. Durch dieses H-Signal hat das Gate von dem Transistor T_1 eine Gate-Spannung und der Transistor T_1 schaltet durch. Damit stellt der Transistor T_1 eine Verbindung zu dem

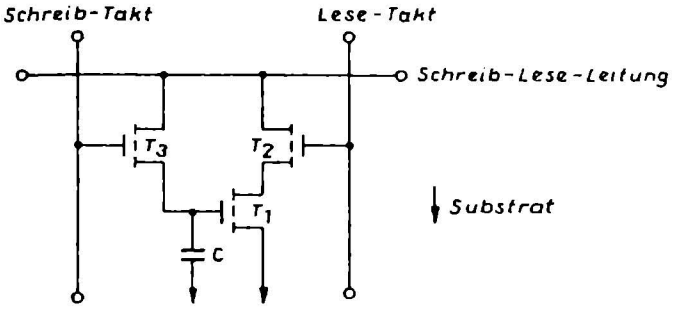


Bild 1.3.3 Aufbau einer dynamischen MOS-Speicherzelle mit drei Transistoren und einem Kondensator

Substrat her. Die Information ist gespeichert, wenn an der Schreib-Takt-Leitung ein L-Signal anliegt.
Soll die gespeicherte Information, also das H-Signal ausgelesen werden, so erhält die Schreib-Takt-Leitung ein H-Signal. Der Transistor T_2 schaltet durch und die Schreib-Lese-Leitung ist mit dem Substrat verbunden. Der Leseverstärker erkennt ein L-Signal, da der Transistor T_1 durchgeschaltet hat. Das L-Signal des Leseverstärkers wird invertiert und liegt mit einem H-Signal an dem Ausgang. Beim Auslesen der gespeicherten Information wird diese nicht zerstört.
Wenn in die Speicherzelle von Bild 1.3.3 ein L-Signal eingeschrieben werden soll, liegt die Schreib-Lese-Leitung auf L-Signal. Durch ein H-Signal auf der Schreib-Takt-Leitung wird der Transistor T_3 leitend und der Kondensator C entlädt sich. Dadurch verringert sich die Gate-Spannung an dem Transistor T_1 und der Kanal wird abgebaut. In der Speicherzelle ist ein L-Signal gespeichert, wenn die Schreib-Takt-Leitung ein L-Signal hat.

Das Auslesen der Information beginnt wieder mit einem H-Signal auf der Lese-Takt-Leitung. Da der Transistor T_1 durch die nicht vorhandene Kondensatorladung gesperrt ist, erkennt der Leseverstärker auf der Schreib-Lese-Leitung ein H-Signal. Dieses H-Signal liegt invertiert an dem Ausgang an.

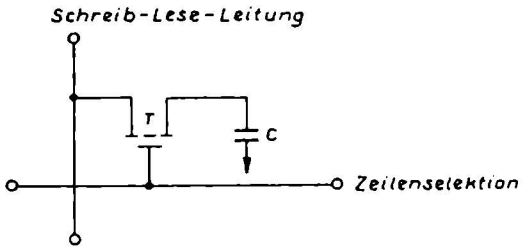


Bild 1.3.4 Aufbau einer dynamischen Ein-Transistor-MOS-Speicherzelle

Die Weiterentwicklung brachte die 1-Transistor-Speicherzelle von Bild 1.3.4. Durch die Zeilenselektion steuern oder adressieren wir die Speicherzelle. Die Daten werden über die Schreib-Lese-Leitung eingeschrieben oder ausgelesen.
In die Speicherzelle von Bild 1.3.4 ist ein H-Signal einzuschreiben. Auf der Schreib-Lese-Leitung befindet sich die H-Information. Durch das H-Signal auf der Selektions-Leitung adressieren wir den Transistor T. Es kann sich ein Kanal aufbauen und der Kondensator lädt sich auf. Die Information ist gespeichert, wenn die Selektions-Leitung ein L-Signal aufweist. Bei einer Speicherung eines L-Signales entlädt sich der Kondensator und hat damit ein L-Signal gespeichert.
Das Auslesen einer Information ist in der Schaltung von Bild 1.3.4 sehr einfach. Ist der Transistor selektiert oder adressiert, besteht zwischen dem Kondensator und der Schreib-Lese-Leitung eine elektrische Verbindung. Ist in der Speicherzelle ein H-Signal gespeichert, so erkennt der Leseverstärker ein H-Signal und bei einem gespeicherten L-Signal ein L-Signal.
Wesentlich an der dynamischen Speicherzelle von Bild 1.3.4 ist die Umschaltung des Schreibverstärkers zum Lese-Verstärker. Beim Schreiben darf der Lese-Verstärker nicht arbeiten und umgekehrt.
In Bild 1.3.5 ist der Aufbau eines dynamischen MOS-RAMs gezeigt. Es handelt sich hier um ein volldecodiertes RAM mit 2048 x 1-Bit. Die Adressierung läuft wie bei dem statischen RAM ab. Die Adressen liegen an, wenn der $\emptyset_1$-Takt ein H-Signal hat. Das gleiche gilt auch für den CS-Eingang. Die Adressierung der 2048 Speicherzellen erfolgt durch 32 Zeilen und 64 Spalten. Mit dem $\emptyset_2$-Takt können

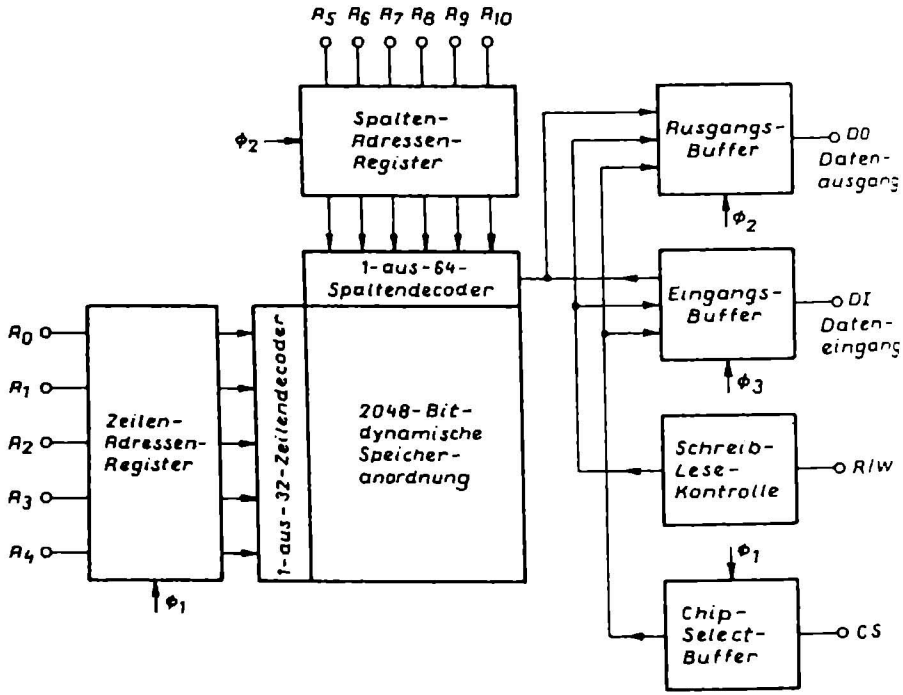


Bild 1.3.5 Aufbau eines dynamischen MOS-RAMs mit der Organisation von 2048x1

die gespeicherten Daten an dem Ausgang DO anstehen, vorausgesetzt, der R/W-Eingang hat ein L-Signal. Informationen werden eingeschrieben, wenn der R/W-Eingang ein H-Signal hat und der $\emptyset_3$-Takt ein H-Signal.
Für das dynamische RAM von Bild 1.3.5 muß extern ein Phasentakt für die drei Taktsysteme erzeugt werden. Hierzu müssen die Zeitintervalle von den einzelnen Herstellern genau beachtet werden, damit eine periodische Auffrischung gewährleistet ist.
Werden keine Daten eingeschrieben oder ausgelesen, so liegt der CS-Eingang auf L-Signal. Damit ist das gesamte RAM blockiert. Während dieser Zeit müssen sich die wertniedrigen Adressen laufend ändern, damit ein Refresh gewährleistet ist. Hierzu liegt an den Adressen-Eingängen A_0 bis A_4 ein 5-Bit-Zähler an, der mit einer Frequenz von 100kHz bis 1MHz arbeiten soll. Bei einer Frequenz von 100kHz wird alle ca. 3ms jede Zeile aufgefrischt. Die Taktsignale $\emptyset_1$, $\emptyset_2$ und $\emptyset_3$ bleiben in dieser Zeit konstant.

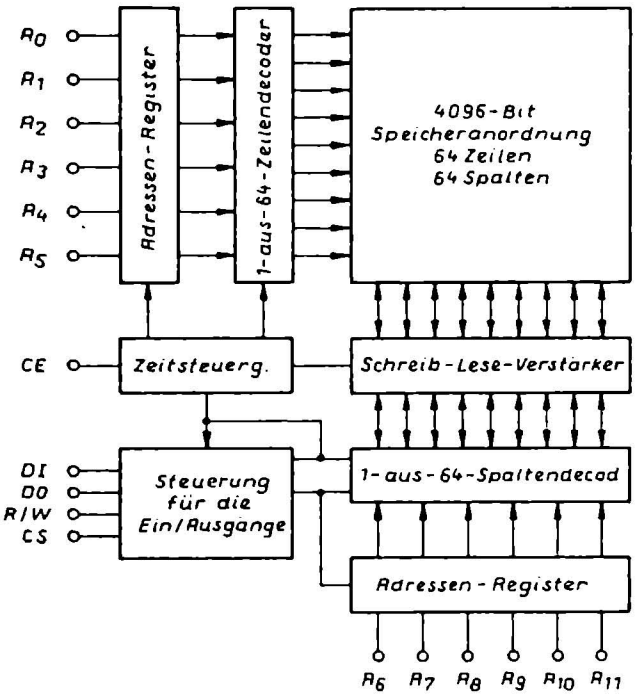


Bild 1.3.6 Dynamisches 4096-Bit-RAM im 22-poligen Gehäuse

In Bild 1.3.6 ist ein anderes dynamisches MOS-RAM gezeigt. Das RAM hat eine Speicherkapazität von 4096 x 1-Bit. Die Informationen in dem RAM werden entweder während eines Lese- oder Schreib-Zyklus jeweils nur auf der ausgewählten Zeilenadresse aufgefrischt. Die Zeilenadresse wird mit den Eingangssignalen A_0 bis A_5 ausgewählt. Jede der Zeilenadressen muß innerhalb eines Zeitraumes von 2ms einen Auffrischzyklus erhalten, da sonst die Daten verloren gehen.

Wenn in dem RAM für das Auffrischen ein Lesezyklus verwendet wird, kann der CS-Eingang ein H- oder L-Signal haben. Wird ein Schreibzyklus verwendet, muß der CS-Eingang ein H-Signal haben.
Das dynamische RAM von Bild 1.3.6 wird in einem 22poligen Gehäuse geliefert. In dem Bild 1.3.7 ist ein dynamisches RAM mit 4098 x 1-Bit gezeigt. Der Aufbau des RAMs ist anders, da dieses RAM in einem 18poligen Gehäuse untergebracht ist.

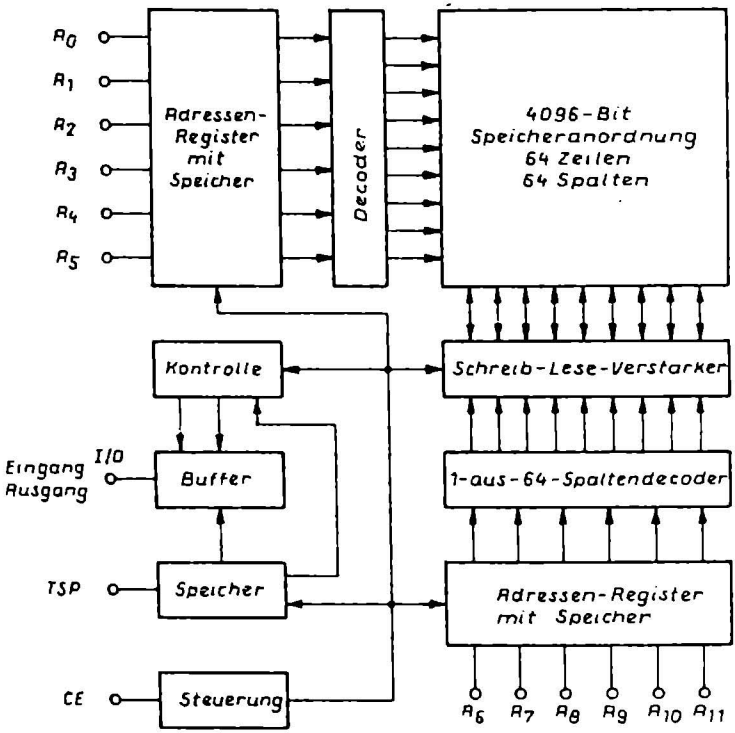


Bild 1.3.7 Dynamisches 4096-Bit-RAM im 18-poligen Gehäuse

Die Ein- und Ausgabe der Daten erfolgt durch den I/O-Buffer. Das Einschreiben und Auslesen der Informationen ist durch den TSP-Steuereingang möglich, (TSP= Tri-Share-Port). Mit einem L-Signal arbeitet der Eingangsverstärker und die Ausgangsstufe ist hochohmig. Bei einem H-Signal arbeitet die Ausgangsstufe und der Eingangsverstärker ist gesperrt. Dies ist nur möglich, da der Eingangsverstärker und die Ausgangsstufe parallel geschaltet sind. Beide Stufen in dem I/O-Buffer sind gesperrt, wenn der CE-Eingang auf L-Signal liegt. Während dieser Zeit können durch die wertniedrigen Adressen A_0 bis A_5 die Zeilen des RAMs aufgefrischt werden.
Für die Auffrischung des RAMs von Bild 1.3.7 ist ein 64-Bit-Zähler erforderlich. Nach ca. 2ms muß jede Zeile aufgefrischt werden. Es ergibt sich daher eine Taktfrequenz von 128kHz. Die Taktfrequenz von 1MHz ist wesentlich günstiger, da die Auffrischung der Speicherzeilen innerhalb von ca. 8ms durchgeführt wird. Dauert ein Einschreibe- oder Auslesevorgang zu lange, so kann durch die höhere Frequenz eine schnellere und damit sichere Auffrischung durchgeführt werden.

1.4. Schreib-Lese-Speicher in CMOS-Technik

Bei der MOS-Technik wird entweder in der n- oder in der p-Kanal-Technik gearbeitet. In der CMOS-Technik wird die n- und die p-Kanal-Technik kombiniert. Durch die Kombination ergeben sich zahlreiche Vorteile, z.B. der geringe Leistungsverbrauch und die Darstellung der Logik:

L-Signal: 0 bis 7,5V

H-Signal. 1,5 bis 15V

Die Darstellung der Logik überschneidet sich. Dies kommt von der Tatsache, daß der Schwellwert zwischen dem L- und H-Signal von der Betriebsspannung abhängig ist. Der Schwellwert zwischen L- und H-Signal liegt auf Höhe der halben Betriebsspannung.

$+U_b = 5V$	L-Signal von 0V bis 2,5V
	H-Signal von 2,5V bis 5V
$+U_b = 10V$:	L-Signal von 0V bis 5V
	H-Signal von 5V bis 10V
$+U_b = 15V$:	L-Signal von 0V bis 7,5V
	H-Signal von 7,5V bis 15V

Ein CMOS-Inverter besteht aus einem p- und einem n-Kanal-MOS-Transistor. In Bild 1.4.1 ist der Aufbau einer CMOS-Speicherzelle gezeigt. Die beiden Transistoren T_2 und T_4 sind p-Kanal-Transistoren. Die anderen sechs Transistoren sind n-Kanal-Typen.

Die eigentliche CMOS-Speicherzelle besteht aus den Transistoren T_2, T_3, T_4 und T_5. Dabei bilden die Transistoren T_2 und T_3, ebenso wie T_4 und T_5 einen Inverter (NICHT-Gatter). Durch die kreuzweise Verbindung der Gate-Anschlüsse ergibt sich eine bistabile Kippstufe, die Informationen speichern kann. Die Ansteuerung verläuft folgendermaßen. Wir geben auf den Punkt A ein L-Signal. Damit schaltet der Transistor T_4 durch und der Transistor T_5 sperrt. Der Transistor T_4 verbindet den Punkt B mit der Betriebsspannung und daher liegt an den Gate-Anschlüssen

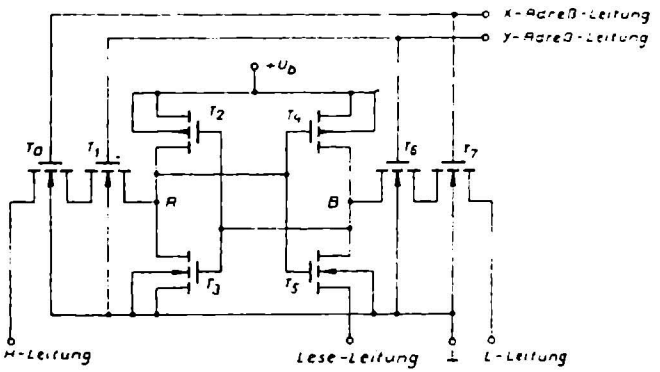


Bild 1.4.1 Aufbau einer CMOS-Speicherzelle

der beiden anderen Transistoren T_2 und T_3 ein H-Signal an. Der Transistor T_2 sperrt und der Transistor T_3 leitet. Der Punkt A bleibt auf L-Signal.
Ändern wir das Signal von Punkt A von L nach H, so erhalten die beiden Transistoren T_4 und T_5 ein H-Signal, d.h. der Transistor T_4 sperrt und der Transistor T_5 schaltet durch. Damit haben wir an dem Punkt B ein L-Signal. Dieses L-Signal liegt durch die kreuzweise Verbindung der Gate-Anschlüsse an dem Gate des Transistors T_2 und T_3 an. Damit ist der Transistor T_2 leitend und der Transistor T_3 sperrt. Der Punkt A liegt auf H-Signal.
Die vier Transistoren T_0, T_1, T_6 und T_7 benötigen wir zur Adressierung der Speicherzelle. Mit einem H-Signal können diese n-Kanal-Transistoren einen Kanal zwischen Source und Drain aufbauen. Der Anschluß von Drain und Source kann hier nicht festgelegt werden, da die beiden Anschlüsse von der Polarität der anliegenden Spannungspotentiale abhängig ist.
Über die X- und Y-Adressen-Leitung wird die Speicherzelle von Bild 1.4.1 angewählt. Mit einem H-Signal auf den Leitungen werden die Transistoren T_0, T_1, T_6 und T_7 leitend. Über die H- oder die L-Leitung kann nun eine Information eingeschrieben werden. Es gilt.

H-Signal: H-Leitung ein H-Signal und L-Leitung ein L-Signal

L-Signal: H-Leitung ein L-Signal und L-Leitung ein H-Signal

Bei der Speicherung eines H-Signales hat der Punkt A ein H-Signal und bei einem L-Signal der Punkt B. Durch die Lese-Leitung kann die Speicherzelle direkt abgefragt werden, ob in der Speicherzelle ein L- oder H-Signal gespeichert ist. Bei einem gespeicherten H-Signal liegt an dem Transistor T_5 ein H-Signal an, d.h. der Transistor hat einen Kanal. Ist dagegen ein L-Signal gespeichert, so ist der Transistor T_5 gesperrt.
Bei den CMOS-RAMs unterscheiden wir zwischen folgenden Typen:

Asynchrones RAM

Ein Ram dieser Bauart ist dadurch gekennzeichnet, daß Änderungen der Adresse sofort weitergegeben werden. Die Adressierung des RAMs erfolgt unmittelbar nach der Änderung.

Synchrones RAM.

Ein RAM nach diesem Prinzip kann seine Adresse erst ändern, wenn ein bestimmter Befehl anliegt. Die Adressierung des RAMs erfolgt nicht unmittelbar, sondern erst nach einem Befehl.

In Bild 1.4.2 ist der Aufbau eines asynchronen RAMs gezeigt. Die Speicher-Matrix besteht aus 256 Speicherzellen, wie sie Bild 1.4.2 zeigt. Die 32 Zeilen der Matrix werden über die fünf Adressen-Eingänge E_0 bis E_4 adressiert. Der Adreß-Inverter erzeugt ein negiertes und ein phasenrichtiges Signal für den Decoder. Der X-Decoder besteht aus 32 UND-Gattern. Durch die Verschaltung der Adreß-Inverter-Ausgänge kann jedes der 32 UND-Gatter angesteuert, also selektiert oder adressiert werden. Die Ansteuerung der acht UND-Gatter in dem Y-Decoder wird durch die drei Adreß-Eingänge E_5, E_6 und E_7 vorgenommen. Die Adressierung wird durch die beiden Decoder sofort an die Speicherzellen weitergegeben.

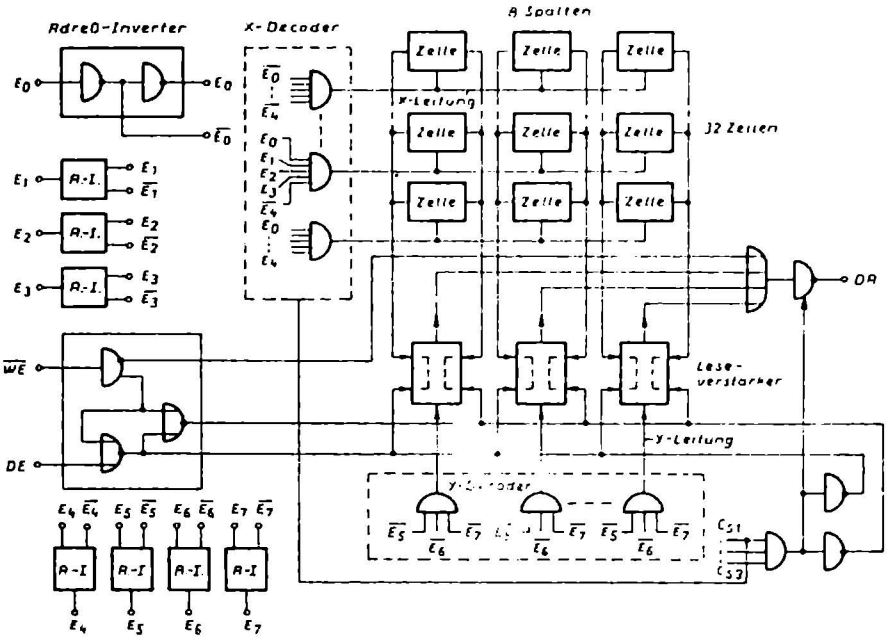


Bild 1.4.2 Asynchrones CMOS-RAM

In Bild 1.4.3 ist der Aufbau eines synchronen RAMs gezeigt. Die Adressen liegen an den Adreß-Registern an. Hat der CS_1-Eingang ein L-Signal, kann die Adressierung direkt erfolgen. Die Register schalten die Adressen sofort durch. Mit einem H-Signal an diesem Eingang wird die Adresse festgehalten. Durch eine Änderung des CS_1-Einganges von H nach L, können wir die Adresse ändern und bei dem L-H-Sprung speichern. Das synchrone RAM hat eine Speicheranordnung von 32 x 32 Speicherplätzen und damit 1024 Speicherplätze.

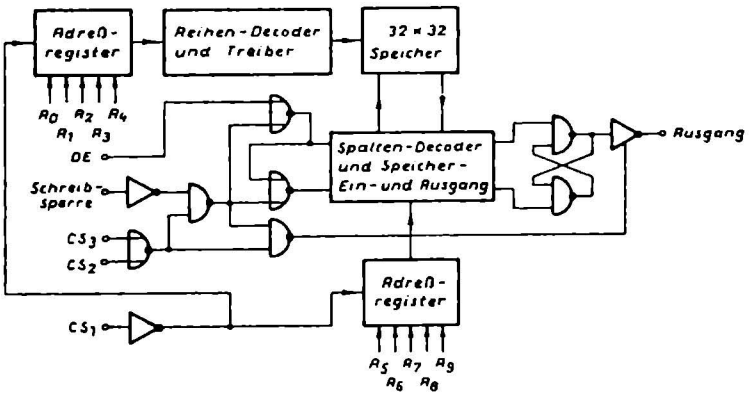


Bild 1.4.3 Synchrones CMOS-RAM

1.5 Festwertspeicher (ROM)

Bei einem Festwertspeicher ROM (read only memory) wird der Speicherinhalt bereits bei der Herstellung eingeschrieben. Die Speicherzellen sind ähnlich wie beim RAM aufgebaut, aber der Speicherinhalt läßt sich nicht mehr ändern. Ein ROM kann daher nur ausgelesen werden. Der Inhalt bleibt in einem ROM erhalten, selbst wenn die Betriebsspannung ausfällt.

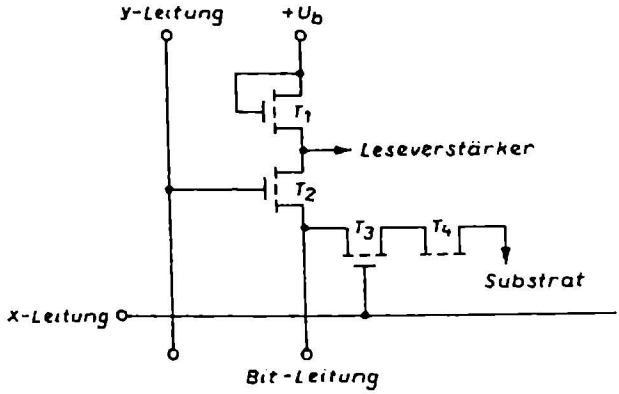
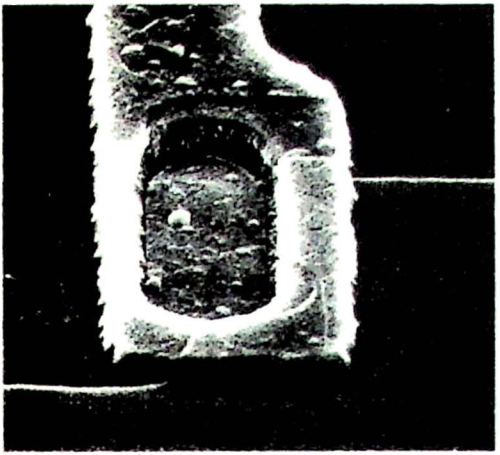


Bild 1.5.1 Aufbau einer ROM-Speicherzelle

In Bild 1.5.1 ist der Aufbau einer ROM-Speicherzelle gezeigt. Die Adressierung der Speicherzelle wird über die beiden x- und y-Leitungen vorgenommen. Die Speicherzelle ist adressiert, wenn an beiden Leitungen ein H-Signal anliegt. Die Transistoren T_2 und T_3 haben daher einen Kanal. Über den Transistor T_1 liegt die Betriebsspannung an dem Leseverstärker an. Dieses H-Signal wird invertiert und daher ergibt sich am Ausgang ein L-Signal. Nach der Adressierung liegt die Bit-Leitung auf L-Signal, wenn der Transistor T_4 leitend ist. Bei einer programmierten Speicherzelle ist das Gate des Transistors T_4 mit der positiven Betriebsspannung verbunden. In Bild 1.5.2 ist diese Kontaktstelle deutlich gezeigt.
Ist der Transistor T_4 in der Schaltung nicht programmiert worden, so hat die Bit-Leitung ein H-Signal, selbst wenn die Speicherzelle adressiert worden ist. In diesem Fall hat der Transistor keinen Kanal zwischen dem Transistor T_3 und dem Substrat.
Durch den Anschluß des Gates von dem Transistor T_4 kann die Speicherzelle programmiert werden. Der Anschluß wird im Maskenverfahren hergestellt. Der Kunde liefert für sein ROM eine Programmiertabelle. Von der Programmiertabelle wird eine Programmiermaske erstellt. Bei jedem Programmierungs-Bit bleibt in der Maske ein Fenster, das Programmierfenster, offen. Durch ein Fotoverfahren wird die Maske auf den Baustein gebracht. Auf der Chip-Oberfläche wird durch das Fotoverfahren alles abgedeckt, bis auf das Fenster. Nach dem Ausätzen der Schicht wird das Gate dieses Transistors mit der Betriebsspannung verbunden. Damit liegt das Gate immer auf H-Signal, d.h. der Kanal ist vorhanden. Bei den anderen Tran-



1.5.2 Kontaktstelle für das Maskenfenster einer ROM-Speicherzelle

sistoren besteht keine elektrische Verbindung von dem Gate zur Betriebsspannung und daher ist auch kein Kanal zwischen Drain und Source vorhanden. Die Speicherzelle ist unprogrammiert.

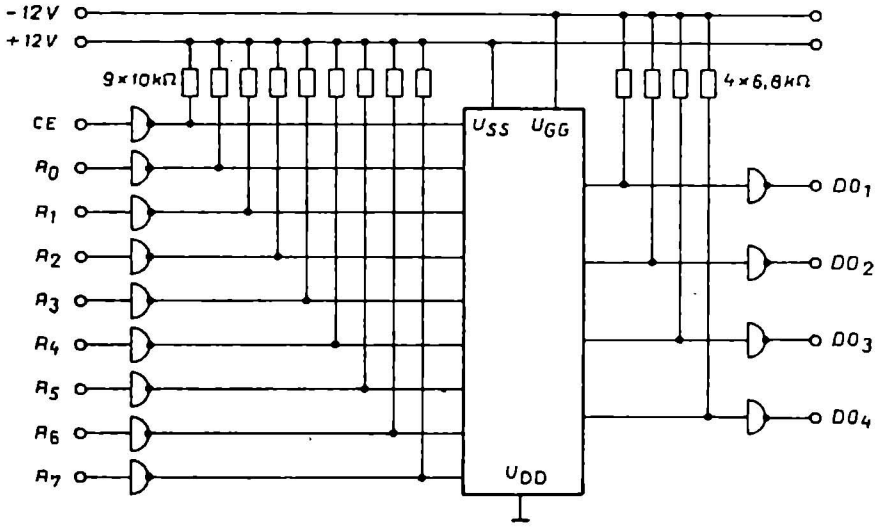


Bild 1.5.3 256x4-Bit-CMOS-ROM mit TTL-Interface

In Bild 1.5.3 ist ein 256 x 4-Bit-MOS-ROM mit dem dazugehörigen TTL-Interface gezeigt. An den Eingängen A_0 bis A_7 liegen die Adressen an. Die TTL-Gatter an den Adressen-Eingängen haben einen offenen Collector und sind über 10kΩ-Widerstände mit der +12V-Betriebsspannung verbunden. Die vier Ausgänge des MOS-ROMs liegen über 6,8kΩ-Widerstände an -12V. Dies ist für den Betrieb erforderlich. Die TTL-Gatter an den Ausgängen bringen den ROM-Signalpegel wieder auf einen TTL-Pegel.

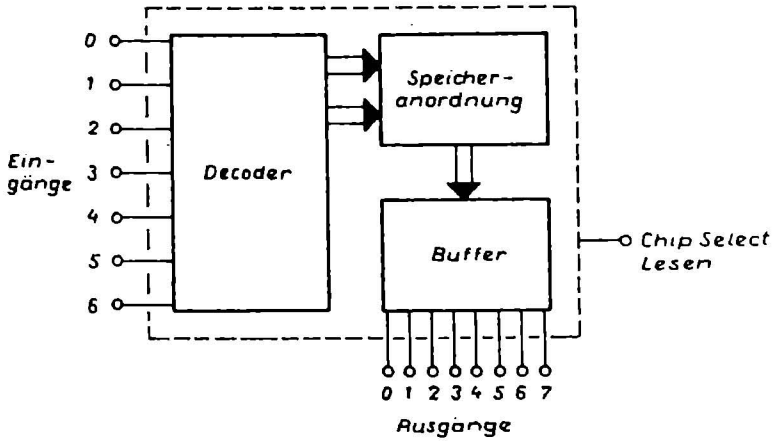


Bild 1.5.4 MOS-ROM mit einer Speicherkapazität von 1024-Bit und einer Organisation von 128x8

Das MOS-ROM von Bild 1.5.4 hat sieben Eingänge und acht Ausgänge. Damit ergibt sich eine Speicherkapazität von 1024-Bits mit einer Organisation von 128 x 8. Die Adressen liegen an dem Eingangs-Decoder an. Die Adressierung der einzelnen ROM-Zeilen erfolgt direkt über 128 Zeilen. Wir haben daher 128 Zeilen mit je 8 Spalten. Dieses Masken-MOS-ROM eignet sich durch den 8-Bit-Ausgang besonders gut für Mikroprozessoren.

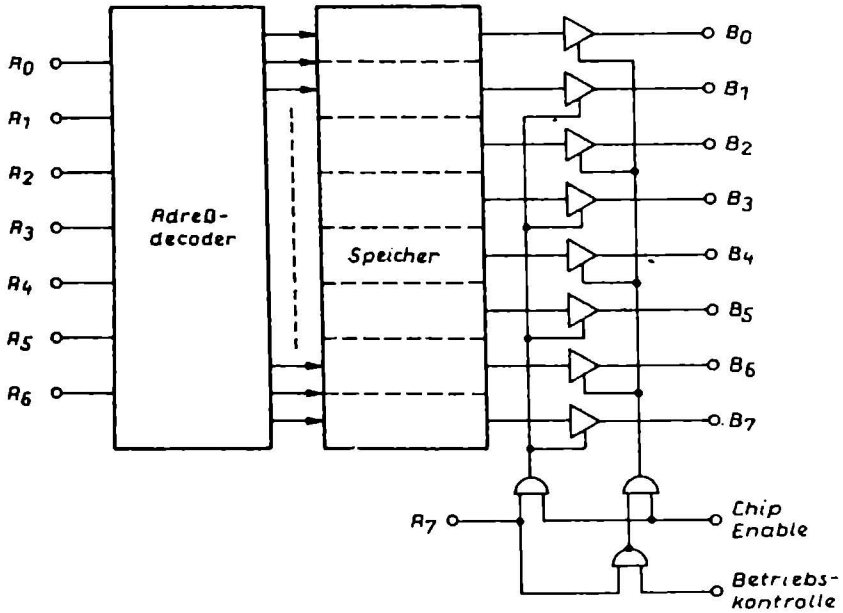


Bild 1.5.5 MOS-ROM mit einer Organisation von 256x8 (2048 Speicherplätze)

Das MOS-ROM von Bild 1.5.5 hat eine Speicherkapazität von 2048 Speicherplätzen. Durch den Adressen-Eingang A_8 können wir dieses ROM in folgenden Organisationen verwenden: 256 x 8 oder 512 x 4.
Die Organisation wird durch den Eingang A_8 bestimmt. Wird der Eingang für eine Adressierung verwendet, so ergeben sich 512 Speicherzeilen. Andernfalls arbeitet die Adressierung nur bis 256 Speicherzeilen. In der 8-Bit-Ausgangsstruktur ist dieses ROM wieder für Mikroprozessoren geeignet.

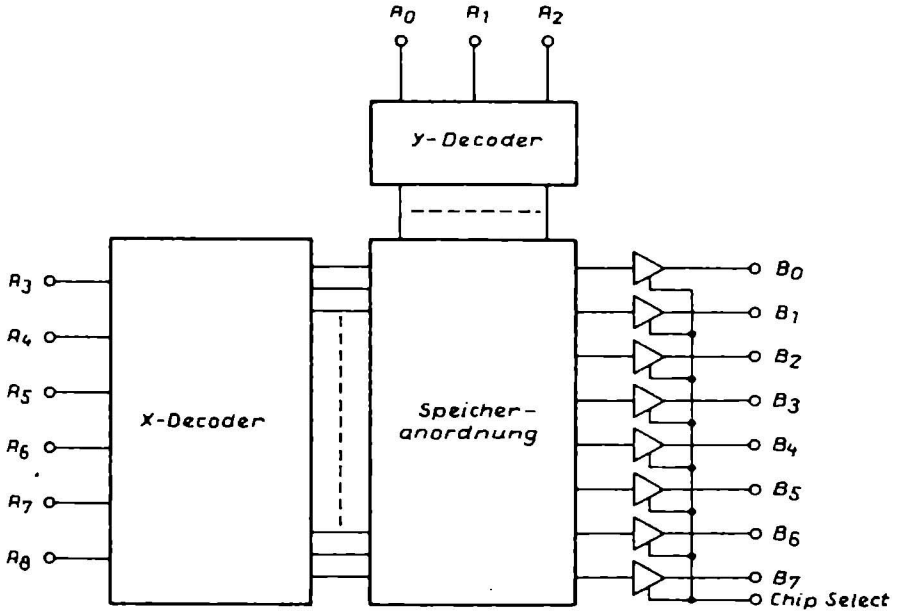


Bild 1.5.6 Mikroprozessor-MOS-ROM mit einer Organisation von 512x8 (4096 Speicherplätze)

Ein weiteres Mikroprozessoren-ROM ist in Bild 1.5.6 gezeigt. Hierbei haben wir eine gesamte Speicherkapazität von 4096-Bits. Für die Adressierung sind die Eingänge von A_0 bis A_8 erforderlich, da das ROM eine Organisation von 512 x 8-Bits hat.

Ein ROM mit zehn Adressen-Eingängen zeigt das Bild 1.5.7. Dieses ROM hat insgesamt 12288 Speicherplätze mit einer Organisation von 1024 x 12 Bits. Dieses ROM kann für 8-Bit- oder 12-Bit-Mikroprozessoren verwendet werden.
Die Zugriffszeit bei diesen MOS-Masken-ROMs liegen in der Größenordnung von 600ns. Wesentlich schneller in der Zugriffszeit sind dagegen die TTL-Masken-ROMs. Hier liegt die Zugriffszeit bei 20ns.
In der Mikroprozessortechnik verwendet man als ROM-MOS-Bausteine die Typen 8308 und 8316. Der Baustein 8308 hat 8192 Bits, mit einer Organisation von 1024 Worte x 8 Bits. Wird ein System entwickelt, so verwendet man das EPROM 8708. Mit diesem EPROM wird der Befehlsinhalt und die Ablaufsteuerung in dem

Mikroprozessor festgelegt. Nach Fertigstellung des Programmes wird das EPROM als Master für die Herstellung einer Maske verwendet. Für die Massenproduktion ist der Baustein 8308, das Masken-ROM, wesentlich billiger, als das EPROM 8708. Beide Bausteine sind pinkompatibel. Dies gilt auch für das MOS-ROM 8316. Der Baustein 8316 hat 16384 Bit mit einer Organisation von 2048-Worten x 8-Bits. Zur Systementwicklung wird aber der Baustein 8716 verwendet. Hierbei handelt es sich ebenfalls um ein EPROM, das wiederum pinkompatibel zum Masken-ROM ist.

Durch die zwei Bausteine 8708 und 8716 kann der Entwickler für den Mikroprozessor sein Programm eingeben oder festlegen. Der Speicherinhalt eines EPROMs kann durch UV-Licht gelöscht werden. Nach dem Löschvorgang wird das neue Programm einprogrammiert. Bei Bedarf kann der Entwickler das EPROM wieder löschen und neu programmieren. Ist der Programmablauf richtig, so dienen diese programmierten EPROMs als Maskenvorlage zur Erstellung der Programmierung für die Masken-ROMs vom Typ 8308 oder 8316. Die Masken-ROMs sind sehr billig in der Herstellung, aber das eingegebene Programm kann nicht mehr geändert werden.

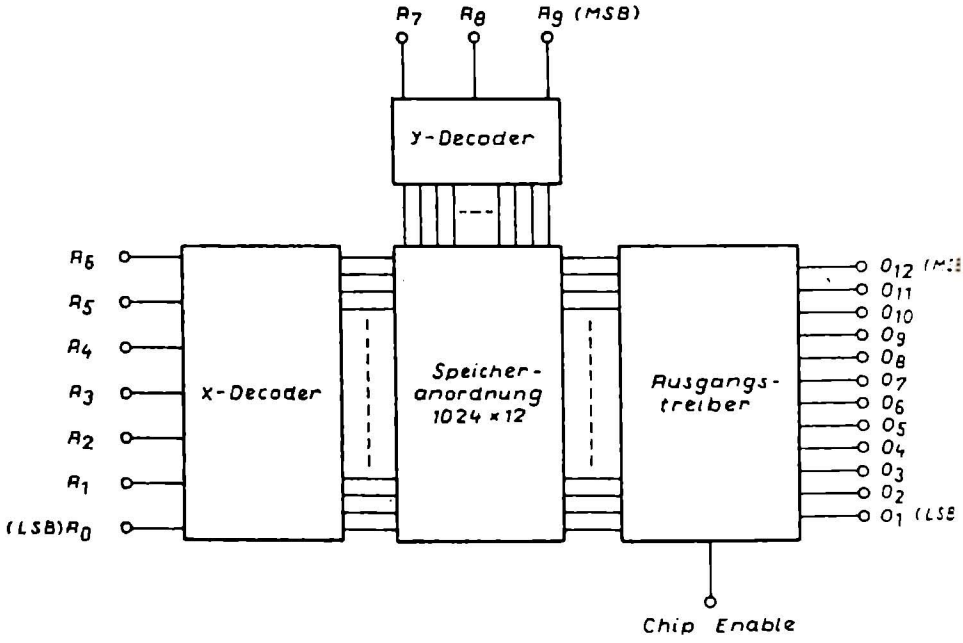


Bild 1.5.7 MOS-ROM mit 12 288 Speicherplätzen, 1024x12

In Bild 1.5.8 ist der Aufbau eines TTL-ROMs gezeigt. Mit den beiden Transistoren T_1 und T_2 kann die eigentliche Programmzelle, die aus dem Transistor T_3 besteht adressiert werden. Die beiden Transistoren T_1 und T_2 schalten durch, wenn an der x- und y-Leitung jeweils ein H-Signal anliegt. Hat die y-Leitung ein H-Signal, schaltet der Transistor T_1 durch und die Bit-Leitung liegt auf H. Der Transistor T_2 schaltet durch, wenn auf der x-Leitung ein H-Signal ist. Damit wird die Programmstelle angesteuert.
Bei einer programmierten Speicherzelle wird die Basis von dem Transistor T_3 mit der Betriebsspannung verbunden. Dadurch ist der Transistor leitend. Im unprogrammierten Zustand ist die Basis nicht angeschlossen.
Hat der Transistor T_1 und T_2 durchgeschaltet, so liegt an der Bit-Leitung ein H-Signal, wenn die Speicherzelle nicht programmiert worden ist. Es kann kein Strom gegen Masse fließen, da der Transistor T_2 gesperrt ist. Bei einem programmierten Zustand fließt ein Strom gegen Masse und das H-Signal auf der Bit-Leitung wird zu L-Signal. Der Lese-Verstärker arbeitet im invertierenden Betrieb, d.h. im nichtprogrammierten Zustand haben wir auf der Bit-Leitung ein H-Signal und daher am Ausgang des Bausteines ein L-Signal. Bei einer programmierten Speicherzelle hat die Bit-Leitung ein L-Signal und daher erscheint am Ausgang des Bausteines ein H-Signal.

In Bild 1.5.8 ist der Aufbau eines Masken-TTL-ROMs gezeigt. Über die Eingänge a bis e erfolgt die Adressierung der einzelnen Zeile. Jeder Eingang ist durch einen einzelnen Inverter gebuffert. Dadurch ergibt sich ein "fan in" von n=1. Die Adressierung kann durch den CE-Eingang blockiert werden. Hat der Eingang ein H-Signal, kann der Baustein nicht arbeiten, da sämtliche UND-Gatter gesperrt sind. Die Ausgänge sind so geschaltet, daß bei einer Gegentakt-Endstufe die unteren Transistoren leitend sind. Bei den Ausgängen mit den offenen Collektoren sind die Transistoren gesperrt. Werden Tri-State-Ausgänge verwendet, sind die Ausgänge hochohmig.

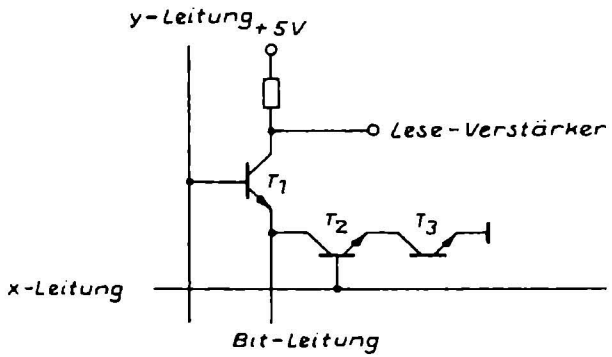


Bild 1.5.8 Aufbau eines TTL-ROMs (Speicherzelle)

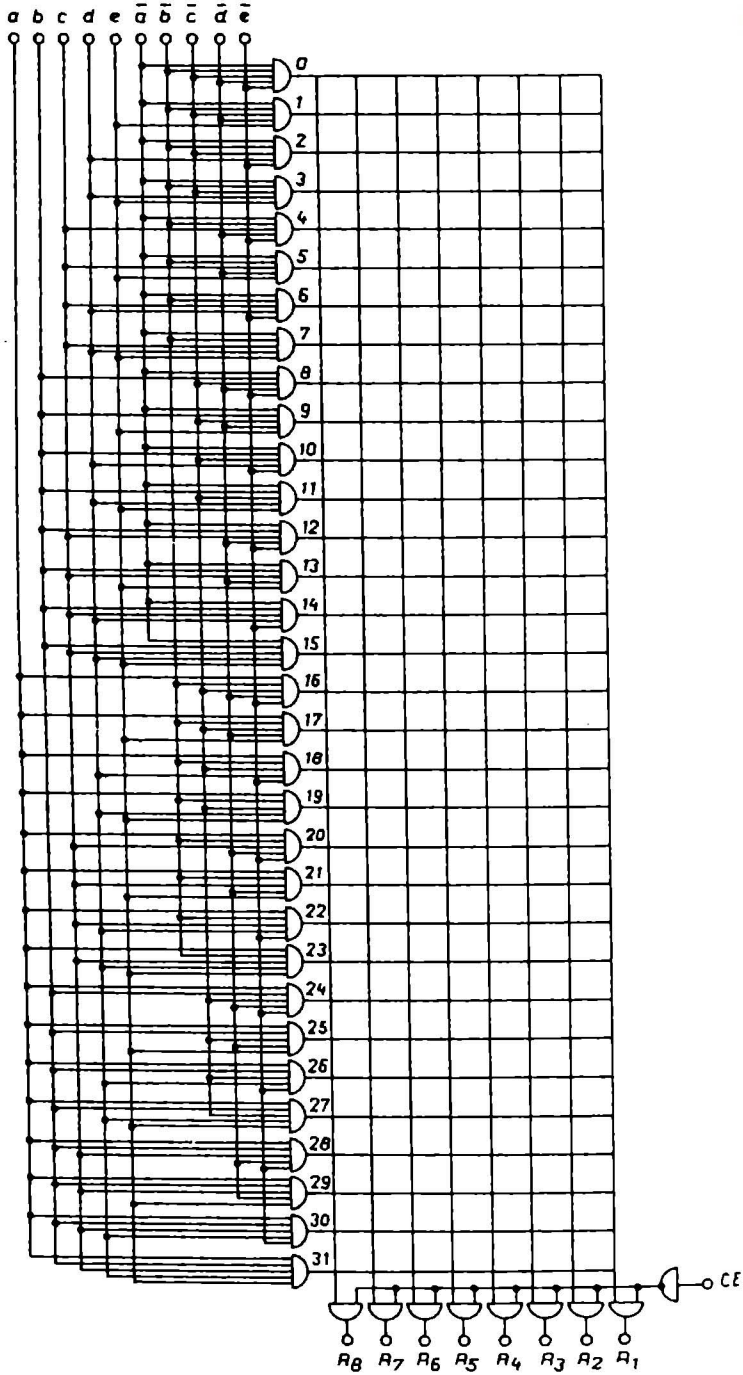


Bild 1.5.8 Aufbau eines TTL-Masken-ROMs (Speichermatrix)

Bei dem ROM in Bild 1.5.8 befinden sich in den Schnitt-Punkten die Transistoren T_3. Beim Programmieren durch die Maske wird hier bei einer Programmstelle ein Fenster offengelassen. Durch ein Fotoverfahren wird die gesamte Oberfläche abgedeckt, bis auf die Programmierfenster. Nach dem Ausätzen der Fenster können die einzelnen Basis-Anschlüsse mit der Betriebsspannung verbunden werden. Damit ist die Programmierung durch das Masken-Verfahren abgeschlossen.
Das ROM von Bild 1.5.8 hat 256 Speicherplätze und diese haben die Organisation von 32-Worten x 8-Bit. Durch die fünf Adressen-Eingänge a bis e kann jede Zeile angesteuert werden. Auf jeder Zeile befinden sich acht Programmierstellen.
In Bild 1.5.9 ist der Aufbau eines 1024-Bit-TTL-ROMs gezeigt. Die Organisation beträgt 256-Wörter x 4-Bit. Die Speicher-Anordnung besteht aus 32 Zeilen und 32 Spalten. Die Spalten sind in 8 x 4-Hauptspalten unterteilt. Dies ist durch die vier Ausgänge bedingt. Mit den Eingängen A, B und C steuern wir den 1- aus 8-Decoder an. Jeder der vier Decoder kann zwischen acht Spalten wählen. Dadurch ergeben sich 32 Spalten in der Speicher-Anordnung. Die Ausgangssteuerung kann durch die G-Eingänge vorgenommen werden. Der Baustein wird mit Gegentakt-, offenem Collektor- oder Tri-State-Ausgang geliefert.

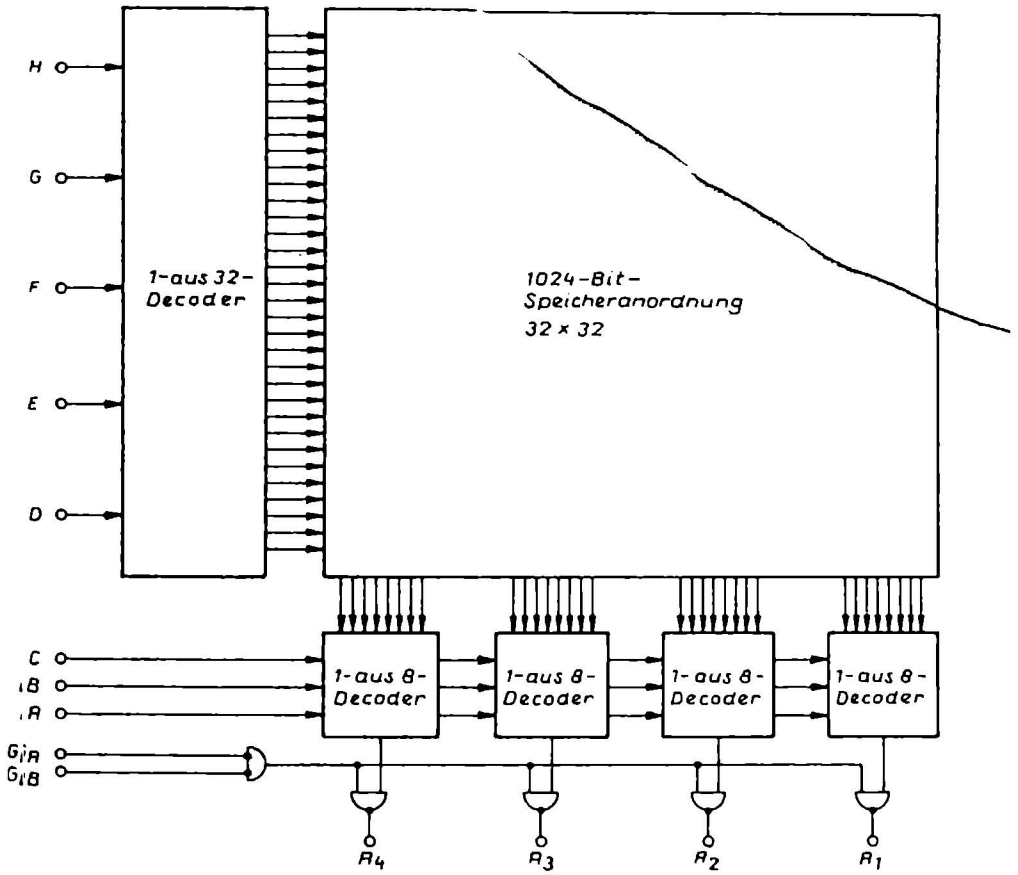


Bild 1.5.9 Aufbau eines 1024-Bit-TTL-ROMs

Ein anderes TTL-ROM ist in Bild 1.5.10 gezeigt. An den Adreß-Eingängen A_0 bis A_7 liegen die Adressen an. Damit ergeben sich 256 mögliche Adressenwörter. Die Speicher-Anordnung besteht aus 256 x 8-Speicherplätzen. Dies ergibt dann 2048 Speicherplätze insgesamt. Die acht Ausgänge liegen an Speicher-Flip-Flops an. Durch die Strobe-Steuerung kann der gespeicherte Inhalt an den Ausgängen festgehalten werden. Während dieser Zeit können sich die Adressen an den A-Eingängen laufend ändern, ohne daß sich der Ausgangszustand ändert. Die beiden CS-Eingänge dienen zur Kapazitätserweiterung.

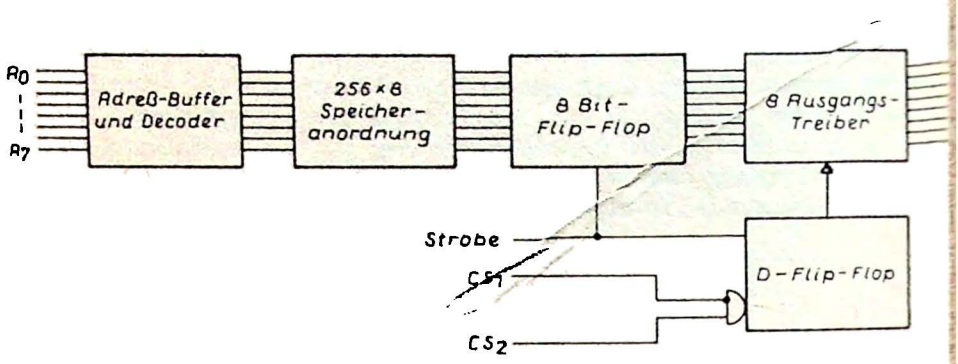


Bild 1.5.10 Aufbau eines 2048-Bit-TTL-ROMs

Durch die Mikroprozessoren wurde es notwendig, die Speicherkapazität der ROMs zu erhöhen. In Bild 1.5.11 ist ein ROM mit der Kapazität von 16384-Bits gezeigt. Die Organisation liegt bei 2048 Wörtern x 8 Bits. Die Speicher-Anordnung selbst besteht aus 128 Zeilen und 128 Spalten. Die Spalten sind in acht Hauptspalten unterteilt. Jede Hauptspalte hat die Informationen für einen Ausgang gespeichert. Damit hat jede Hauptspalte 16 Unterspalten. Diese Unterspalten werden durch den 1- aus 16-Decoder adressiert. Die Zugriffszeit der Informationen liegt in der Größenordnung von 50ns.

Seit kurzem sind auf dem Halbleitersektor die ersten CMOS-ROMs erhältlich. Bild 1.5.12 ist der Aufbau eines CMOS-ROMs gezeigt. Im wesentlichen ist das CMOS-ROM wie die MOS-ROMs aufgebaut. Nur werden hierbei statt der MOS-Transistorwiderstände die p-Kanal-Transistoren verwendet. Damit kann die Betriebsspannung für ein CMOS-ROM wieder von 5V bis 15V reichen.

Unter die Gruppe der ROMs fallen auch die Charaktergeneratoren. In Bild 1.5.13 ist die gesamte Schaltung für einen Charaktergenerator gezeigt. An den Adreß-Eingängen A_1 bis A_6 liegen die Daten-Eingänge des ASCII-Codes an. Durch den Eingang-Buffer (7416, NICHT-Gatter mit offenem Collektor) werden die Eingangs-Informationen auf +12V angehoben. Die Informationen steuern das ROM an. Das ROM wurde bereits von der Halbleiter-Firma für diese Decodierung programmiert. Die Programmierung setzt der ASCII-Code so um, daß die Eingangsinformationen sichtbar werden. Es gilt folgende Funktionstabelle:

A_6	A_5	A_4	A_3	A_2	A_1	Anzeige	A_6	A_5	A_4	A_3	A_2	A_1	Anzeige
L	L	L	L	L	L	@	H	L	L	L	L	L	
L	L	L	L	L	H	A	H	L	L	L	L	H	!
L	L	L	L	H	L	B	H	L	L	L	H	L	"
L	L	L	L	H	H	C	H	L	L	L	H	H	#
L	L	L	H	L	L	D	H	L	L	H	L	L	$
L	L	L	H	L	H	E	H	L	L	H	L	H	%
L	L	L	H	H	L	F	H	L	L	H	H	L	&
L	L	L	H	H	H	G	H	L	L	H	H	H	'
L	L	H	L	L	L	H	H	L	H	L	L	L	(
L	L	H	L	L	H	I	H	L	H	L	L	H	)
L	L	H	L	H	L	J	H	L	H	L	H	L	*
L	L	H	L	H	H	K	H	L	H	L	H	H	+
L	L	H	H	L	L	L	H	L	H	H	L	L	,
L	L	H	H	L	H	M	H	L	H	H	L	H	-
L	L	H	H	H	L	N	H	L	H	H	H	L	.
L	L	H	H	H	H	O	H	L	H	H	H	H	/
L	H	L	L	L	L	P	H	H	L	L	L	L	0
L	H	L	L	L	H	Q	H	H	L	L	L	H	1
L	H	L	L	H	L	R	H	H	L	L	H	L	2
L	H	L	L	H	H	S	H	H	L	L	H	H	3
L	H	L	H	L	L	T	H	H	L	H	L	L	4
L	H	L	H	L	H	U	H	H	L	H	L	H	5
L	H	L	H	H	L	V	H	H	L	H	H	L	6
L	H	L	H	H	H	W	H	H	L	H	H	H	7
L	H	H	L	L	L	X	H	H	H	L	L	L	8
L	H	H	L	L	H	Y	H	H	H	L	L	H	9
L	H	H	L	H	L	Z	H	H	H	L	H	L	:
L	H	H	L	H	H	[	H	H	H	L	H	H	;
L	H	H	H	L	L	\	H	H	H	H	L	L	<
L	H	H	H	L	H	]	H	H	H	H	L	H	=
L	H	H	H	H	L	(	H	H	H	H	H	L	>
L	H	H	H	H	H	←	H	H	H	H	H	H	?

Funktions-Tabelle für einen Charaktergenerator

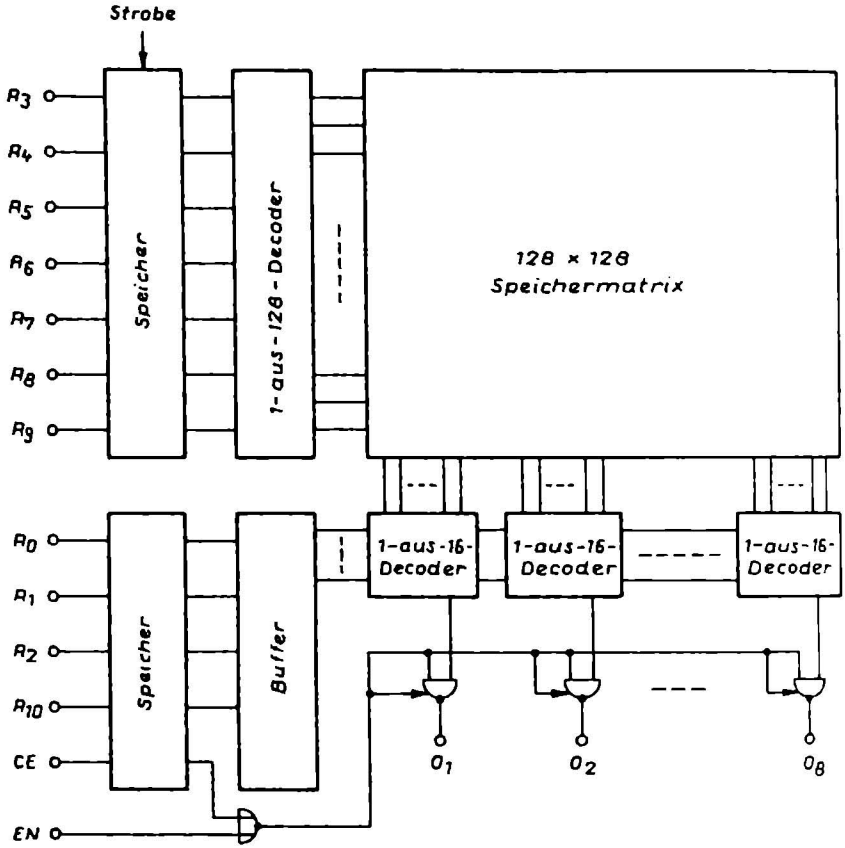


Bild 1.5.11 MOS-ROM mit einer Speicherkapazität von 16 384 Bits

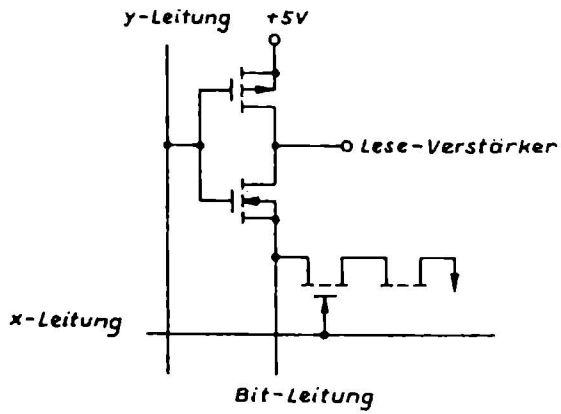


Bild 1.5.12 Aufbau eines CMOS-ROMs

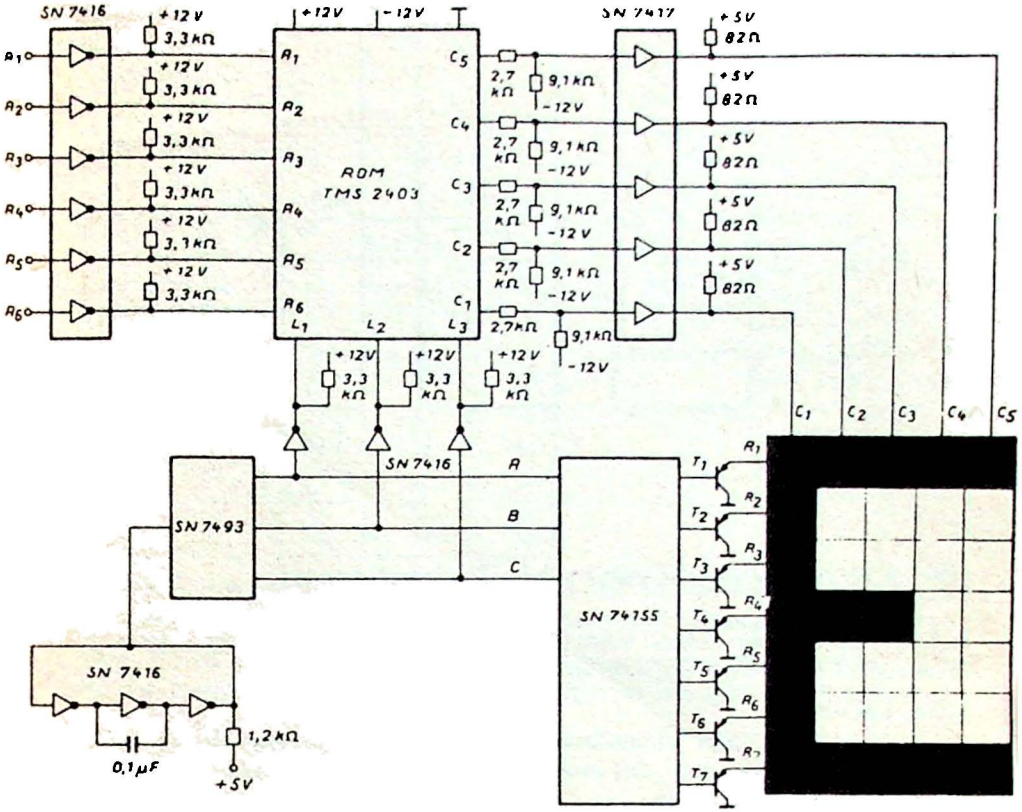


Bild 1.5.13 Schaltung eines Charaktergenerators

Die Zeichendarstellung erfolgt auf einer 5 x 7-LED-Punktmatrizen-Anzeige. Diese Anzeigen werden von folgenden Firmen hergestellt:

Litronix:	DL 57
Monsanto:	MAN 2
Texas Instruments:	TIL 305

An den Eingängen A_1 bis A_6 liegen die Eingangsinformationen an. Gesteuert wird die gesamte Schaltung von dem Oszillator, der aus dem Baustein 7416 besteht. Die Oszillatorfrequenz liegt an dem 3-Bit-Binärzähler an. Mit den drei Ausgängen steuert der Baustein 7493 das ROM und den Decoder 74155. Bei dem Zählerstand 0 ist das ROM blockiert und der Baustein 74155 steuert die 0-Zeile , die aber nicht vorhanden ist. Wir haben hier die Synchronisierstelle. Hierbei kann sich das Datenwort an den Eingängen ändern. Mit dem ersten Takt erhalten wir den Zählerstand 1. Das ROM gibt decodiert die Eingangsinformation auf die Spalte C_1 bis C_5. Der Transistor T_1 erhält ein L-Signal und schaltet durch. Die erste Zeile kann damit leuchten. Mit dem Zählerstand 2 wird der Transistor T_2 durch ein L-Signal leitend und die Zeile A_2 liegt auf Masse. Durch die ROM-Decodierung leuchtet in diesem Fall nur der Leuchtpunkt oder die Leuchtdiode C_1/A_2 auf.

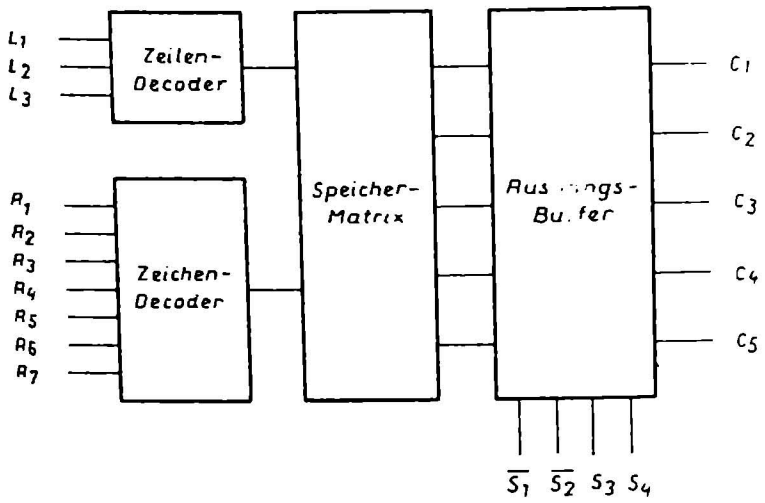


Bild 1.5.14 Aufbau eines Charakters- oder Zeichengenerators

In Bild 1.5.14 ist der Aufbau eines Charakter- oder Zeichengenerators gezeigt. Im wesentlichen besteht der Baustein aus vier Funktionsblöcken:

Zeilen-Decoder.

In dem Zeilen- oder Linien-Decoder werden die einzelnen Zeilen der Speicher-Matrix angewählt. Die Ansteuerung des Zeilen-Decoders erfolgt durch den externen 3-Bit-Zähler.

Zeichen-Decoder:

Der Zeichen-Decoder decodiert die sieben Eingangsinformationen des ASCII-Codes. Die Ausgänge des Zeichen-Decoders sind ebenfalls mit der Speicher-Matrix verbunden.

Speicher-Matrix:

Die Speichermatrix ist ein festverdrahtetes ROM, das von dem Hersteller festgelegt ist. Die Eingangsdaten von dem Zeichen-Decoder ändern sich nicht. Die Steuerung erfolgt von dem Zeilen-Decoder her.

Ausgangs-Buffer.

In den Ausgangs-Buffer werden die Ausgangssignale der Speicher-Matrix verstärkt. Die fünf Ausgänge C_1 bis C_5 steuern die Spalten der Anzeigeneinheit an. Die vier Steuereingänge S sind Sperr-Eingänge.

In dem Bild 1.5.13 ist der Buchstabe E gezeigt. Dies ergibt folgende Funktionstabelle:

	Zeilen			Spalten				
Nr.	L_3	L_2	L_1	C_1	C_2	C_3	C_4	C_5
0	L	L	L	L	L	L	L	L
1	L	L	H	H	H	H	H	H
2	L	H	L	H	L	L	L	L
3	L	H	H	H	L	L	L	L
4	H	L	L	H	H	H	L	L
5	H	L	H	H	L	L	L	L
6	H	H	L	H	L	L	L	L
7	H	H	H	H	H	H	H	H

Synchronisierungs-Zeile

Durch die Zeilensteuerung ergeben sich die einzelnen Funktionen bei den Spalten. Hat der Baustein 7493 den Zählerstand 0, so wird keine Anzeige durchgeführt.

In Bild 1.5.15 ist der Aufbau eines Charaktergenerators der Firma Monolithic Memories gezeigt. An den Eingängen des Bausteines MM 6061 liegen die Eingangs-

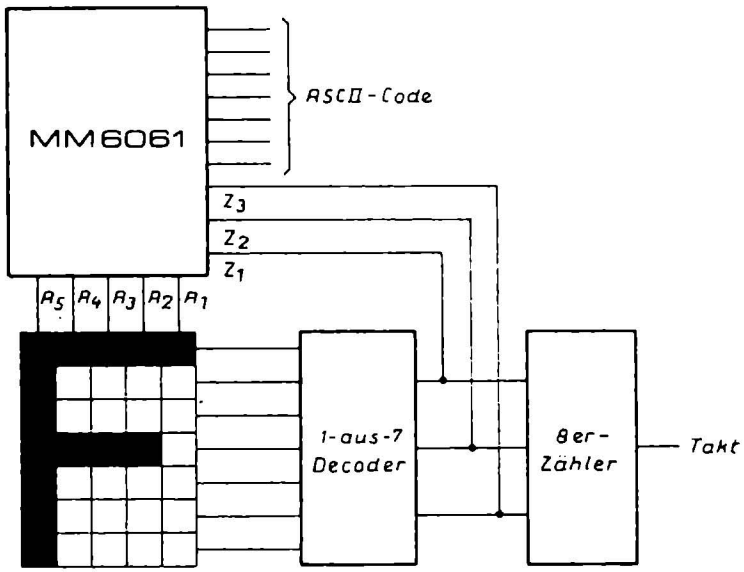


Bild 1.5.15 Aufbau des Charaktergenerators MM 6061

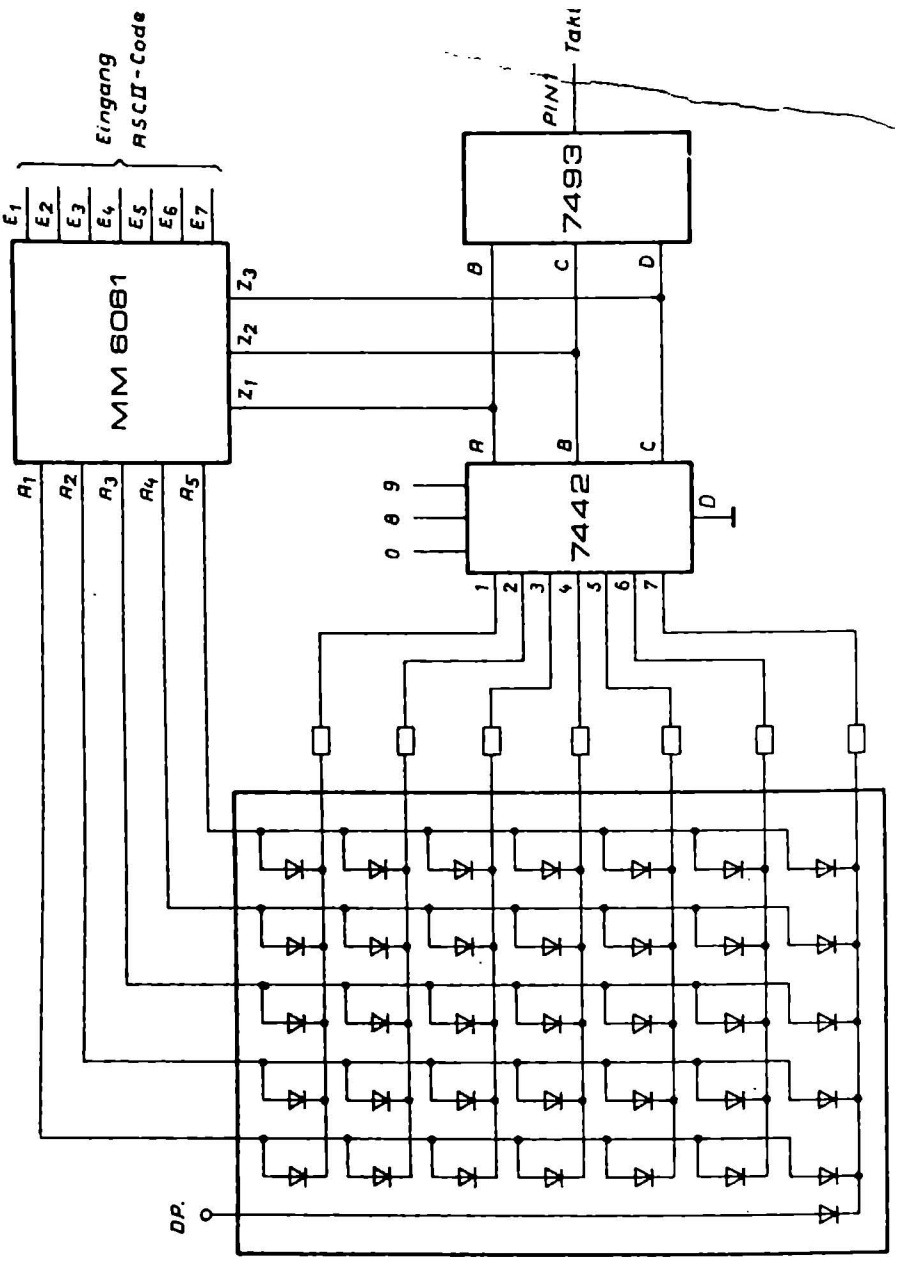


Bild 1.5.16 Schaltung eines Charaktergenerators mit 5 x 7 -LED-Anzeige

informationen des ASCII-Codes an. Mit dem Takt-Eingang wird ein 8er-Zähler angesteuert. Der Zähler zählt von 0 bis 7 und muß sich danach wieder auf 0 zurücksetzen. Die drei Ausgänge des Zählers steuern die Zeilen-Eingänge Z oder L des Bausteines 6061 an und den 1- aus 7-Decoder. Verwenden wir hierzu den Baustein 7442 als Decoder, so sind keine Treiber-Transistoren erforderlich. Die Ausgänge des Bausteines 6061 steuern die fünf Spalten der Anzeige an. Die Zeilen der Anzeige werden durch den Decoder-Baustein angesteuert.
Der Zähler zählt von 0 bis 7. Bei dem Zählerstand 0 wird keine Anzeige durchgeführt. Diesen Zählerstand können wir als Synchronisier-Eingang verwenden. Hat der Zähler diesen Zählerstand, so können wir die Eingangsinformation des ASCII-Codes ändern. Nach diesem Zählerstand ist der Eingang blockiert. Bei dem Zählerstand1 decodiert der Baustein 6061 den ASCII-Code so, daß bei dem Buchstaben F alle fünf Ausgänge ein H-Signal aufweisen. Der 1- aus 7-Decoder hat bei diesem Zählerstand ein L-Signal. Damit wird die erste Zeile auf H-Signal gelegt und es kann ein Strom über die Leuchtdioden fließen.
Beim nächsten Takt schaltet der Ausgang 2 des Decoders die zweite Zeile auf ein L-Signal. Bei dieser Zeile erzeugt der Charakter-Generator nur an dem Ausgang A_5 ein H-Signal und nur eine Diode leuchtet auf.
Für den Baustein 6061 ergibt sich bei der Darstellung des Buchstabens F folgende Funktionstabelle:

	Zeilen			Spalten				
	Z_3	Z_2	Z_1	A_5	A_4	A_3	A_2	A_1
0	L	L	L	L	L	L	L	L
1	L	L	H	H⁻	H⁻	H⁻	H⁻	H
2	L	H	L	H	L	L	L	L
3	L	H	H	H	L	L	L	L
4	H	L	L	H⁻	H⁻	H⁻	H	L
5	H	L	H	H	L	L	L	L
6	H	H	L	H	L	L	L	L
7	H	H	H	H	L	L	L	L

Der dritte Taktimpuls ändert den Zählerstand erneut. Es erfolgt eine Weiterschaltung auf die dritte Zeile. Die Änderung der Zeilen und der Spalten geschieht mittels der Taktfrequenz. Nach dem 8. Taktimpuls springt der Zähler von 7 nach 0 und es erfolgt keine Zeilen-Ansteuerung der Anzeige.

In Bild 1.5.16 liegt an dem B-Eingang des Zähler-Bausteines 7493 die Taktfrequenz an. Die drei Ausgänge des Zählers liegen an den A, B und C-Eingängen des Decoder-Bausteines 7442 an. Wichtig bei dem Decoder ist, daß der D-Eingang unbedingt mit Masse verbunden sein muß. Ist dies nicht der Fall, kommt es zu einer Fehldecodierung.
Die Ausgänge des Zähler-Bausteines sind mit den Eingängen des Zeilen-Decoders 6061 verbunden. Die Ausgänge des Charakter-Generators steuern die einzelnen Spalten der Leucht-Dioden-Matrix an. Die Möglichkeiten der Zeichendarstellung sind in Bild 1.5.17 gezeigt.

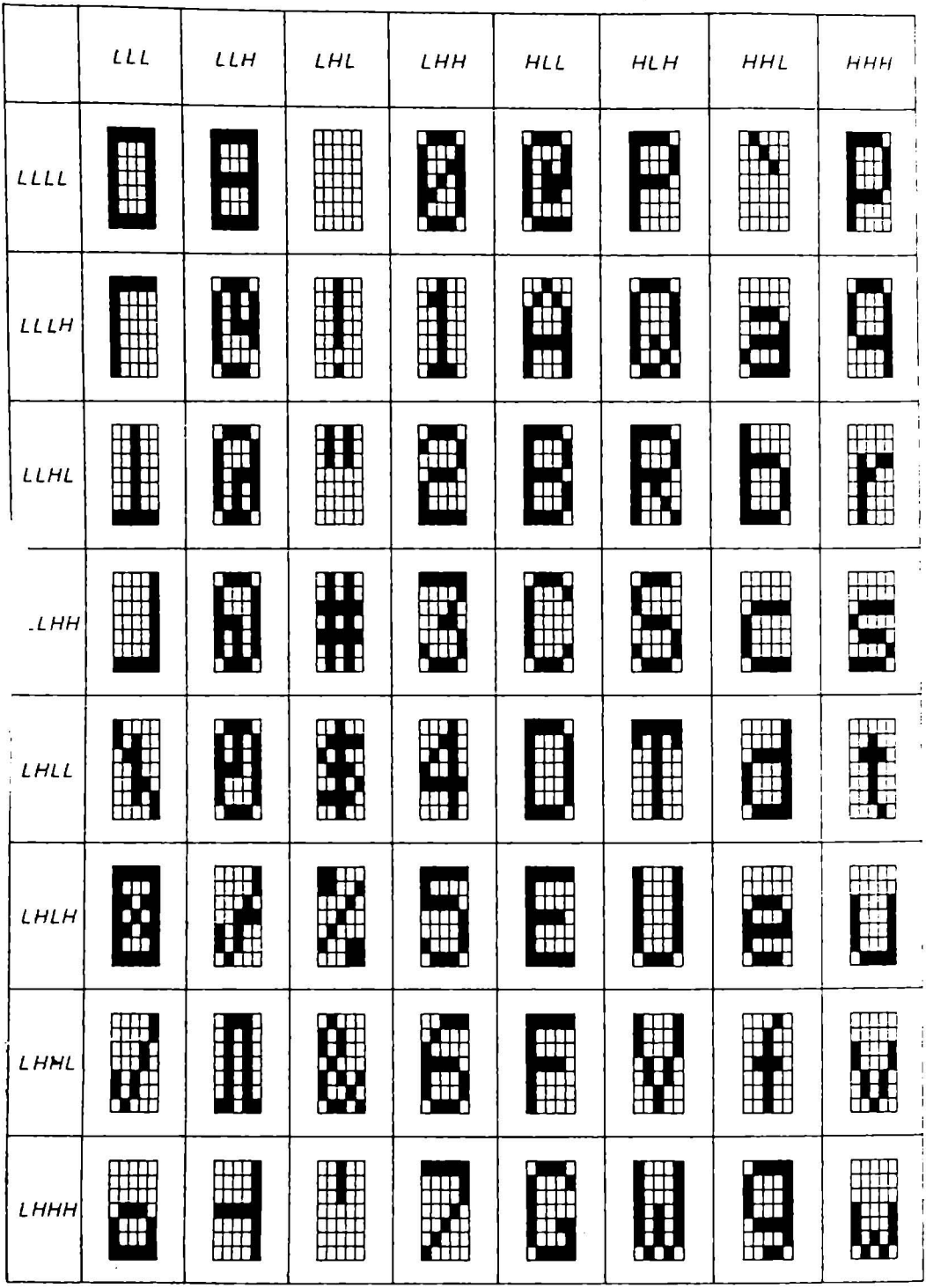


Bild 1.5.17.1 Zeichendarstellung

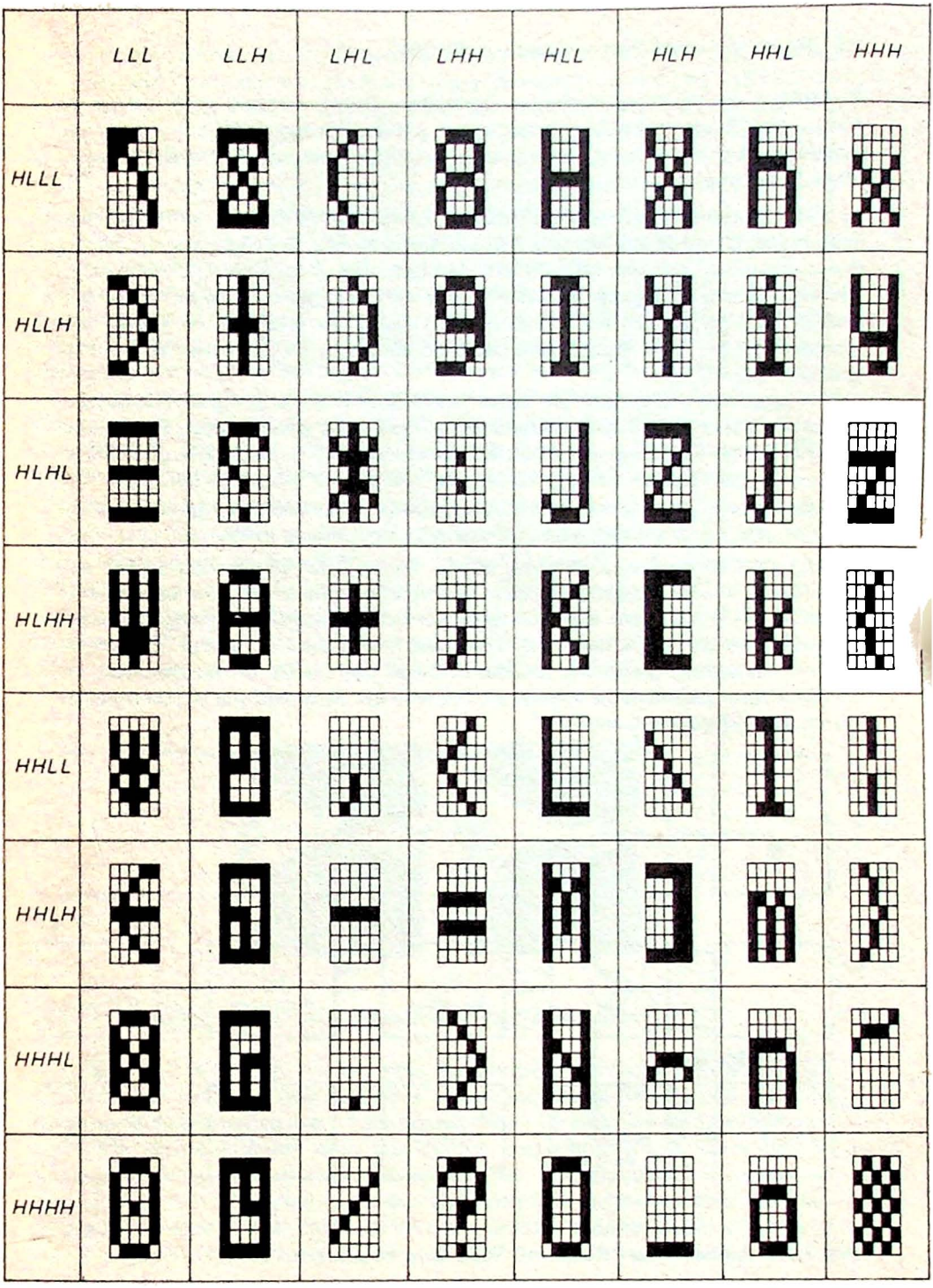


Bild 1.5.17.2 Zeichendarstellung

1.6. Programmierbare Festwertspeicher (PROM)

Ein PROM ist ein programmierbarer Speicher. Durch Anlegen einer Spannung kann in den Baustein ein Programm eingeschrieben werden. Dieses Programmeinschreiben kann direkt vom Anwender durchgeführt werden. Er benötigt hierzu einen ROM-Programmierer.

In Bild 1.6.1 ist der Aufbau eines PROMs gezeigt. Das PROM hat 1024 Speicherplätze und die Organisation von 256 Worten x 4 Bit. Im wesentlichen ist der Funktionsaufbau mit den MOS-ROMs identisch. Die Adressierung erfolgt durch die acht Adressen-Eingänge A_0 bis A_7. Die wertniedrigeren Adressen steuern in diesem Fall die Spalten an. Durch die vier Ausgänge erhalten wir wieder vier Hauptspalten. In jeder Hauptspalte befinden sich acht Unterspalten für die Programmierung.
Die Programmierung können wir selbst durchführen. Im unprogrammierten Zustand hat jede der 1024 Speicherzellen eine intakte Sicherung (fuse). Damit ergeben sich in der Schaltung von Bild 1.6.1 insgesamt 1024 Sicherungen. Jede Sicherung stellt eine leitende Verbindung zwischen der x- und y-Leitung dar. Beim Programmieren wird durch einen Stromstoß die Sicherung zerstört und damit besteht zwischen der x- und y-Leitung keine elektrische Verbindung mehr.

In Bild 1.6.1 ist die Programmierung für das "fuse" oder Sicherungselement eingezeichnet. An den Ausgang O_4 legen wir eine Spannung von 10V an. Die Programmierung erfolgt,wenn der E_1-Eingang auf ein bestimmtes Signal gelegt wird. Dadurch kann ein Strom durch den Transistor in den Decoder fließen. Der Strom passiert das Sicherungselement, die Matrix-Diode und die Diode des Decoders. Da das Speicherungselement die schwächste Stelle in der Stromleitung ist, schmilzt sie durch. Die Zelle ist programmiert.

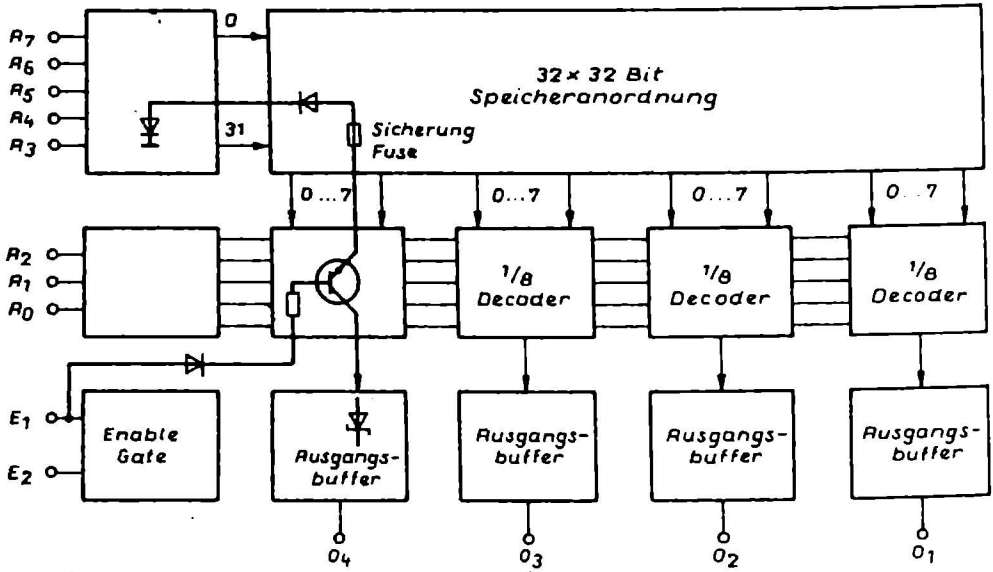


Bild 1.6.1 Aufbau eines PROMs mit 1024 Speicherplätzen

In der Praxis verläuft der Vorgang etwas anders. Wir adressieren durch die Eingänge zuerst den Speicherplatz. Nach der abgeschlossenen Adressierung wird die Programmierspannung an den Ausgang gelegt. Die Programmierung erfolgt, wenn der E_1-Eingang auf H-Signal gelegt wird. Es kann ein Programmierstrom fließen, der das Speicherungselement abschmilzt. Danach sperren wir durch ein L-Signal den E_1-Eingang. Die Programmierung ist abgeschlossen. Die Programmierspannung kann auf einen anderen Ausgang gelegt werden, z.B. auf den Ausgang O_1. Liegt die Spannung an, geben wir auf den Eingang E_1 ein Signal und eine andere Speicherzelle wird programmiert.
Die Programmierdauer für ein Sicherungselement dauert ca. 1ms bei den richtigen Programmierspezifikationen. Zur Zeit werden für das Sicherungselement verschiedene Materialien verwendet. Dadurch sind bei den einzelnen Herstellern verschiedene Programmierspezifikationen üblich. Bevor ein PROM programmiert wird, sind daher unbedingt diese Spezifikationen beim Hersteller einzuholen. Der andere Weg für das Programmieren von PROMs sind die ROM-Programmer der einschlägigen Industrie. Hier wird dann ein Grundgerät geliefert. Je nach Hersteller-ROM wird bei der Programmierung eine andere Programmer-Karte verwendet.

In Bild 1.6.2 ist ein Foto über eine programmierte und eine unprogrammierte Speicherzelle gezeigt. Das Sicherungselement in dieser Speicherzelle besteht aus einem Nickeldraht. In dem Bild rechts ist der Nickeldraht noch unprogrammiert, d.h. wir haben eine elektrische Verbindung. Durch den Programmierstrom wird der Draht abgeschmolzen. Die Schmelztemperatur liegt bei den meisten ROMs zwischen 400°C und 800°C. Dies ist vom Material der Sicherungselemente abhängig.

Bild 1.6.2 Programmierte und unprogrammierte Speicherzelle

Das Bild 1.6.3 zeigt den Aufbau einer Speicherzelle mit Sicherungselement. Die Bit-Leitungen bestehen hier aus einer Aluminium-Metallisierung. Unterhalb der Bit-Leitung befindet sich die x-Leitung oder Wort-Leitung. Die Bit-Leitungen sind gegen die Wort-Leitungen elektrisch isoliert angebracht. Durch die beiden Sicherungselemente wird die Verbindung von den Bit-Leitungen zu den Wort-Leitungen hergestellt. Durch die p- und n-Zone zwischen der Basis und der Wort-Leitung entsteht eine Diode. Durch diese Diode erfolgt eine Signaltrennung von der Bit-Leitung zur Wort-Leitung.

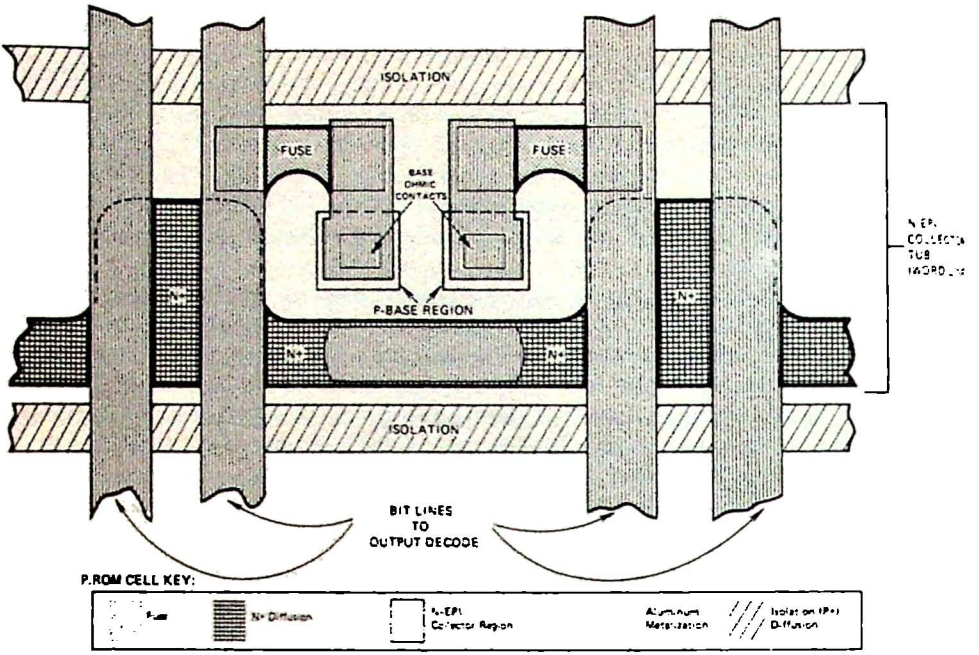


Bild 1.6.3 Aufbau einer Speicherzelle mit Sicherungselement

Bei der Programmierung werden die Sicherungselemente zerstört, d.h. das Sicherungselement wird abgebaut. Wir können uns diesen Vorgang auch an einer Haushaltssicherung vorstellen. Ist die Zerstörung richtig durchgeführt, so können die Sicherungselemente nicht mehr zusammenwachsen. Bei einer unsachgemäßen Programmierung kann es durch einen Wärmeschock zu einer elektrischen Verbindung innerhalb des Sicherungselementes kommen. Dies ist möglich, wenn die Programmierung zu niedrig und zu kurz war. Das Sicherungselement wird in diesem Fall nicht richtig zerstört und es bleiben Materialreste zwischen den Sicherungsenden zurück. Bei einem Wärmeschock kommt es in dem Spalt zu einer Materialwanderung und der Kanal wächst zu.

In Bild 1.6.4 ist der Querschnitt einer ordnungsgemäßen programmierten Speicherzelle gezeigt. Zwischen den beiden Aluminiumblöcken befindet sich das Sicherungselemt. Beim Programmieren verbrennt das Material und die Materialreste oxidieren. Der Nickel-Chrom-Widerstand ist 175 Å dick. Es wird zum Programmieren ein Strom von ca. 400mA benötigt.

Das Diagramm von Bild 1.6.5 zeigt die Programmierung. Die Programmierlinie A zeigt in der Mitte einen steilen Anstieg, wenn richtig programmiert wird. Durch die hohe Wärme in der Mitte des Sicherungselementes erfolgt eine richtige Durchschmelzung des Drahtes. Die Programmierlinie B zeigt die verkehrte Programmierspannung. Durch eine verminderte Leistungszufuhr beim Programmieren

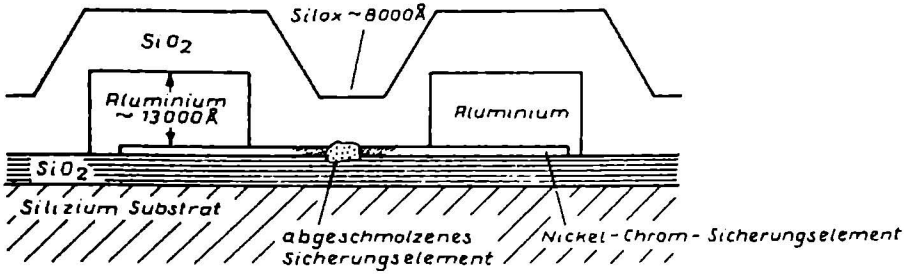


Bild 1.6.4 Querschnitt einer programmierten Speicherzelle

kommt es nicht richtig zum Abschmelzen. Daher entsteht an dem Sicherungselement ein Wärmestau, aber dieser Wärmestau bezieht sich auf das gesamte Sicherungselement. Bei der Kurve A kommt es durch die richtige Programmdosierung nur in der Mitte zum Wärmestau. Daher erfolgt genau in der Mitte eine ordnungsgemäße Trennung. Bei der Kurve B wird die Trennung an mehreren Stellen durchgeführt. Das Material schrumpft nicht, sondern es wird porös. Nach dem Programmieren scheint bei der Kurve B eine richtige Programmierung durchgeführt worden zu sein. Im Laufe der Zeit und unter Wärmeeinfluß kommt es zum Zusammenwachsen (growback) der beiden Enden. Die programmierte Speicherzelle hat sich damit regeneriert.

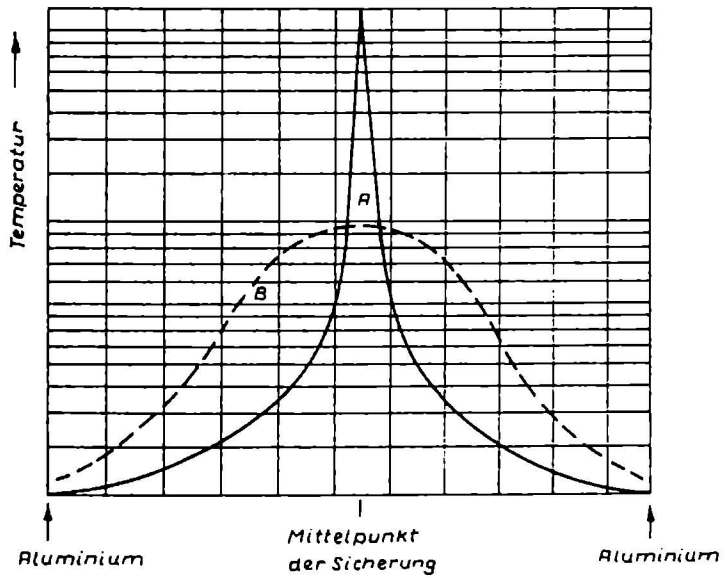


Bild 1.6.5 Programmier-Diagramm

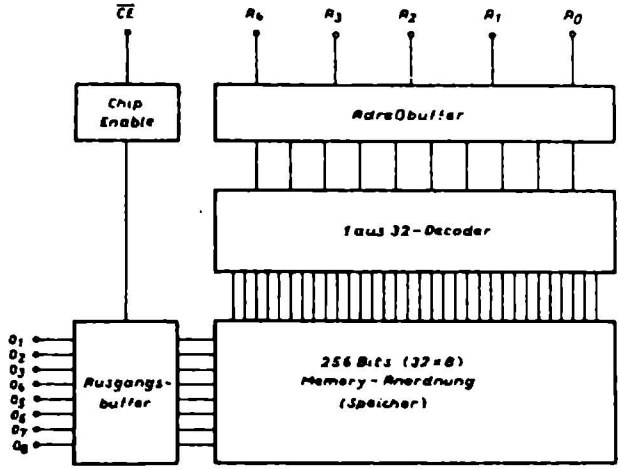


Bild 1.6.6 Aufbau eines 256-Bit-TTL-PROMs

In Bild 1.6.6 ist der Aufbau eines 256-Bit-TTL-ROMs gezeigt. Im wesentlichen sind diese Festwertspeicher mit den TTL-ROMs identisch. Mit den fünf Eingängen werden 32 Zeilen (Wort-Leitungen) adressiert. Die Bit-Leitungen sind mit den acht Ausgängen verbunden. Bei einer Programmierung werden die Sicherungselemente zwischen der Wort- und Bit-Leitung durch einen Programmierstrom entfernt.

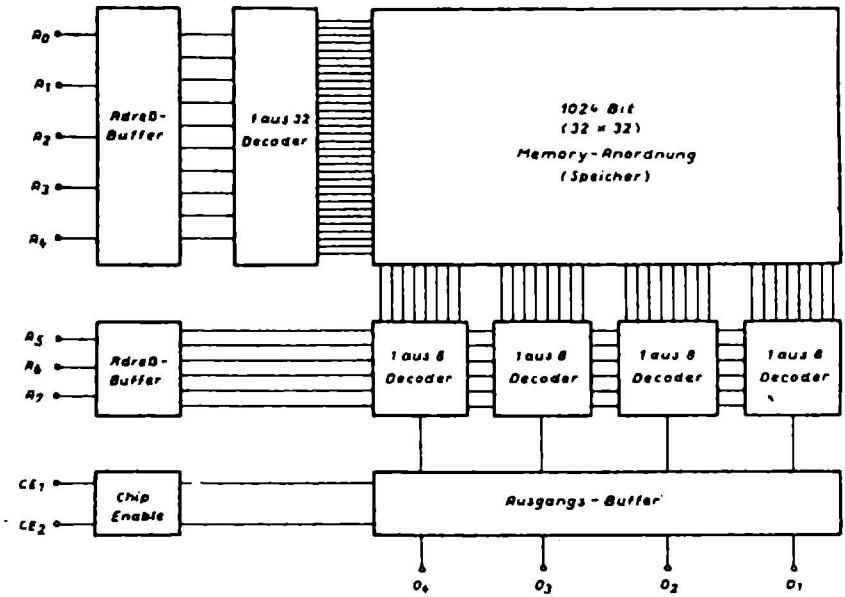


Bild 1.6.7 Aufbau eines 1024-Bit-TTL-PROMs

Das PROM von Bild 1.6.7 ist ein 1024-Bit-TTL-ROM. Der Aufbau ist mit dem PROM von Bild 1.6.1 identisch.

Ein weiteres PROM ist in Bild 1.6.8 gezeigt. Hierbei handelt es sich um ein 2048-Bit-PROM mit der Organisation von 512 x 4-Bits. Bei diesem PROM hat jede Hauptspalte 16 Unterspalten. Die Adressierung der Unterspalten wird durch die Adressen A_5, A_6, A_7 und A_8 bestimmt. Die vier Eingänge steuern die vier internen 1- aus 16-Decoder an.

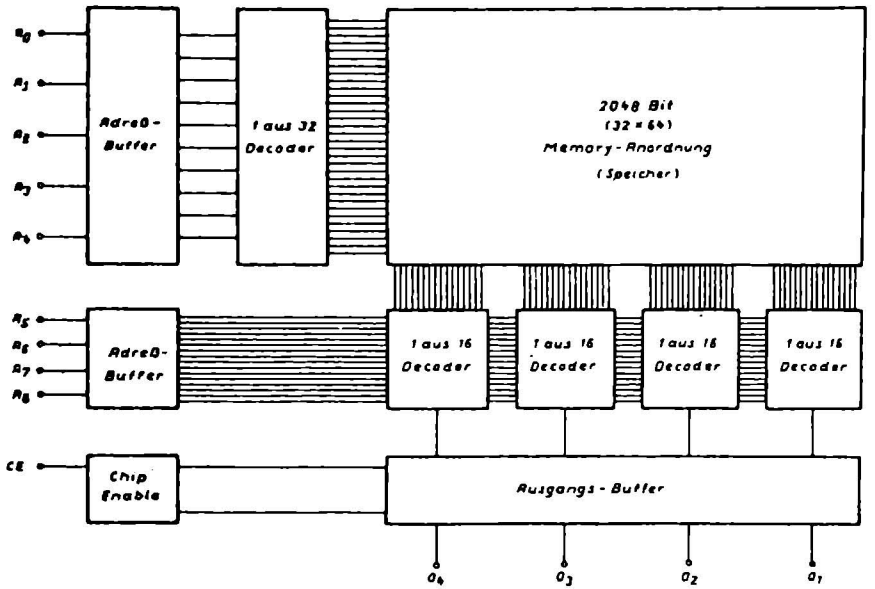


Bild 1.6.8 Aufbau eines 2048-Bit-TTL-PROMs

Im Kapitel 1.5 wurden bereits die EPROMs erwähnt. In Bild 1.6.9 ist der Aufbau einer löschbaren Speicherzelle gezeigt. Die Speicherzelle besteht aus einem einzigen MOS-Transistor. Beim MOS-Transistor besteht das Gate im Gegensatz zu den üblichen MOS-Transistoren aus zwei Schichten. Hierbei handelt es sich um zwei Poly-Silizium-Schichten. Die obere Schicht in Bild 1.6.9 dient als Gate-Anschluß und ist mit der Zeilen-Leitung (x-Leitung) elektrisch verbunden. Die darunter liegende zweite Schicht speichert die Ladung. Durch die Injektion von energiereichen Elektronen in dieser Schicht wird die Speicherzelle aktiviert, d.h. programmiert. Die Elektronen bleiben in dieser Schicht, dem zweiten Gate gespeichert. Die Ladungen können nur durch UV-Licht gelöscht werden. Da zwischen dem zweiten Gate keine leitende Verbindung zu den anderen Funktionen besteht, bleiben die Ladungen gespeichert. Durch diese gespeicherten Ladungen verändert sich die Schwellwertspannung beträchtlich. Im unprogrammierten Zustand liegt der Schwellwert bei 1,5V bis 1,8V. Beim Programmieren erhöht sich dieser Wert auf 3,5V bis 4,2V. Vor der Programmierung hat daher die Speicherzelle eine niedrige

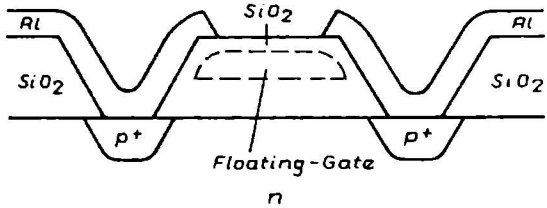


Bild 1.6.9 Aufbau einer EPROM-Speicherzelle

Schwellwertspannung. Durch die Ladung der zweiten Gate-Schicht erhöht sich der Schwellwert. Dies hat zur Folge, daß die Spannung auf der Zeilen-Leitung nicht mehr ausreicht zur Öffnung des Transistors.
Die Programmierung von EPROMs wird durch die Anhebung des Schwellwertes des Transistors programmiert. Das Laden der zweiten Gate-Schicht dauert ca. 0,1s. Nach dieser Zeit ist die Zelle programmiert. Wichtig ist noch, daß der Lese-Verstärker eine ungeladene Gate-Schicht als H-Signal und eine geladene Schicht als L-Signal erkennt. Das zweite Gate wird auch als "floating-gate" bezeichnet.
Beim Lesen der Speicherzelle wird festgestellt,indem wir den Zustand der Spannung überprüfen. Befindet sich die Zelle im ungeladenen Zustand, bringt die Anwahl durch die Wort-Leitung einen wesentlich höheren Stromfluß zwischen Source und Drain. Ist die Zelle geladen und erhält das erste Gate ein H-Signal, so kann fast kein Strom zwischen Source und Drain fließen. Die Ladung in dem zweiten Gate verhindert die Wirkung des elektrischen Feldes auf den Kanal. Im geladenen Zustand kann sich zwischen Source und Drain kein Kanal aufbauen, da die Ladung des zweiten Gates dem elektrischen Feld des ersten Gates entgegenwirkt.

Das Programmieren von dem zweiten Gate ist etwas umständlich gegenüber den anderen PROMs. Während beim PROM durch einen Stromimpuls das Sicherungselement durchgeschossen wird, muß die Ladung im zweiten Gate das EPROM langsam aufgebaut werden, d.h. während des Programmierens muß das Programmiergerät ca. 100 Programmschleifen durchlaufen. Die Programmierschleife beginnt bei der Adresse 0 und dem Ausgang 1. Danach wird der Ausgang 2, 3, 4 usw. an die Programmierspannung angelegt. Ist die erste Zeile durch, wird die Adresse auf 1 erhöht. Danach wird Ausgang für Ausgang an die Programmierspannung gelegt. Die Dauer des Programmierimpulses liegt bei 1ms. Hat das EPROM eine Speicherkapazität von 1024 Bits, so dauert ein Programmierumlauf ca. 1s. Nach dem Programmierumlauf beginnt der Programmer wieder von vorne, bei der Adresse 0 (Zeile 0) und dem Ausgang 1. Der Programmiervorgang dauert daher ca. 100 Sekunden. Innerhalb dieser Zeit wurde jede Zelle, die programmiert werden soll, mit 100 Programmierimpulsen angesteuert. Nach dieser Zeit ist ein EPROM richtig programmiert. Erfolgt die Programmierung einer Speicherzelle konstant, so ist der Baustein innerhalb kurzer Zeit zerstört, da durch die energiereichen Elektronen an dem zweiten Gate ein zu großer Wärmestau entstanden ist. Durch die schrittweise Programmierung erhöht sich die Programmierbarkeit wesentlich, von ca. 10 Programmierungen auf ca. 10^5. Der Programmierstrom soll bei der Schleifenprogrammierung ca. 20mA betragen.

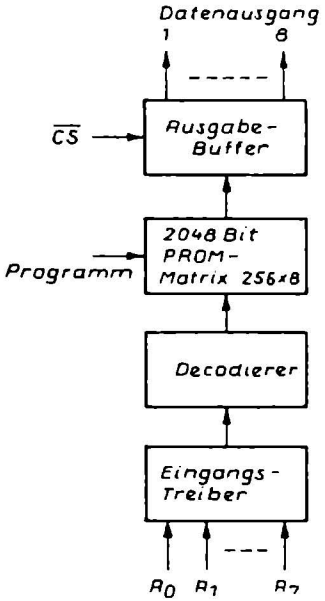


Bild 1.6.10 Aufbau eines 2048-Bit-EPROMs (8702)

In Bild 1.6.10 ist der löschbare und elektrisch programmierbare Festwertspeicher 8702 gezeigt. Der Baustein hat 2048 Bit mit einer 256 x 8-Bit-Organisation. Der Baustein ist direkt pinkompatibel zu dem Festwertspeicher 8302. Dies gilt auch für die anderen technischen Daten. An den Adreß-Eingängen A_0 bis A_7 liegen die Adressen-Bytes an. Durch diese Eingangsstruktur kann der Baustein direkt von dem 8080 betrieben werden. Die Eingangssignale werden in dem Decoder decodiert und anschließend steuert der Decoder die PROM-Matrix an. Die acht Bit-Leitungen sind über den Ausgabe-Buffer mit den Ausgängen verbunden.
Nach einer ordnungsgemäßen Programmierung der einzelnen Speicherzellen bleibt der Inhalt unbegrenzt erhalten. Die Programmierung dauert ca. 2 Minuten für alle 2048 Bits. Durch die statische Technologie ist kein Auffrischungs-Zyklus erforderlich.
Die PROM-Matrix in dem 8702 befindet sich unter einem Quarzfenster. Dadurch kann der gesamte Speicherinhalt durch die Bestrahlung von ultraviolettem Licht vom Entwickler gelöscht und danach wieder elektrisch programmiert werden. Dieser Vorgang kann beliebig oft wiederholt werden, sofern die Programmierung in einem vorhin beschriebenen Zyklus durchgeführt wird. Für den 8702 ergeben sich folgende Programmier-Anweisungen:

Programmieren:

Zu Beginn sind alle 2048 Bits des EPROMs im L-Zustand. Die Informationen werden eingegeben durch das selektierte Programmieren von H-Zuständen an den gewünschten Speicherstellen.
Die Adressen-Anwahl wird durch die gleiche Adressierung, wie beim Lesen durchgeführt. Alle acht Eingänge müssen sich auf umgekehrtem (negiertem)

Zustand befinden, wenn die getakteten Spannungen auf ihre negative Spannungen umschalten. Die Adressen sollten bis 8702 mindestens 32mal in der Folge von 0 bis 255 angesprochen sein. Die Programmiergeräte-Hersteller lassen diese Folge ca. 100mal durchlaufen. Damit ist eine dauerhafte Programmierung gewährleistet. Eine Daten-Eingangsspannung von -48V hinterläßt ein L-Signal in der PROM-Matrix. Alle acht Bit eines Wortes (Zeile) werden gleichzeitig programmiert, indem die gewünschten Bit-Informationen auf den Daten-Eingabe-Eingängen eingestellt werden.

Löschvorgang.

Bei den EPROMs kann immer nur der gesamte Speicherinhalt durch das Quarzfenster gelöscht werden. Hierzu verwenden wir ultraviolette Strahler in einer Wellenlänge von 2537 Å. Die UV-Lampen dürfen keine Filter haben und sind in einem Abstand von 3 cm aufzustellen. Achtung! Durch das starke UV-Licht werden die menschlichen Augen blind !!! Innerhalb von 20 Minuten ist der gesamte Speicherinhalt gelöscht.

Bei größeren Speicherkapazitäten verwenden wir den EPROM-Typ 8708 mit einer Speicherkapazität von 8192 Bits und einer 1024 x 8-Bit-Organisation. In Bild 1.6.11 ist der Aufbau gezeigt. Die Eingänge A_0 bis A_9 sind die Adressen-Eingänge, die Ausgänge sind mit O_1 bis O_8 gekennzeichnet und der $\overline{CS}$/WE-Eingang dient zur Baustein-Auswahl/Schreib-Freigabe. Der Baustein ist pinkompatibel zu dem Festwertspeicher 8308. (Masken-ROM).
Die 8k-Speicherplätze können innerhalb von zwei Minuten programmiert werden. Für die Programmierspezifikationen können die Werte des 8702 verwendet werden. Da dieses EPROM statisch arbeitet, ist kein Auffrisch-Zyklus für die Speicherzellen erforderlich.

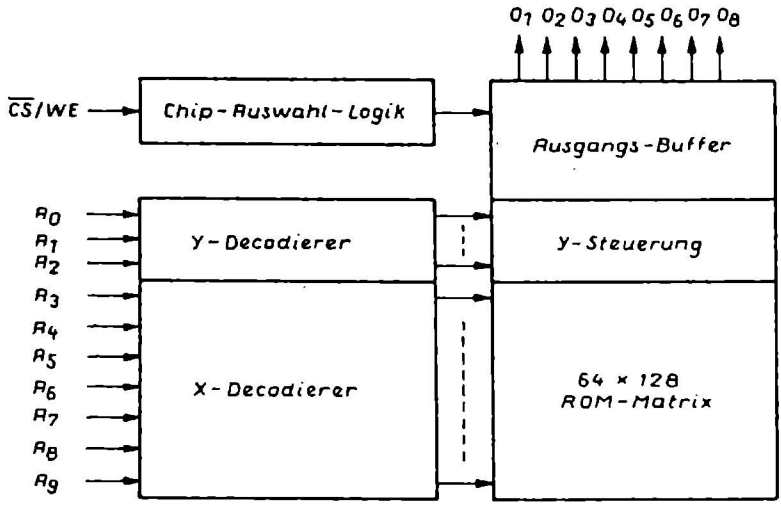


Bild 1.6.11 Aufbau des EPROMs 8708 (8192-Bits, 1024 x 8)

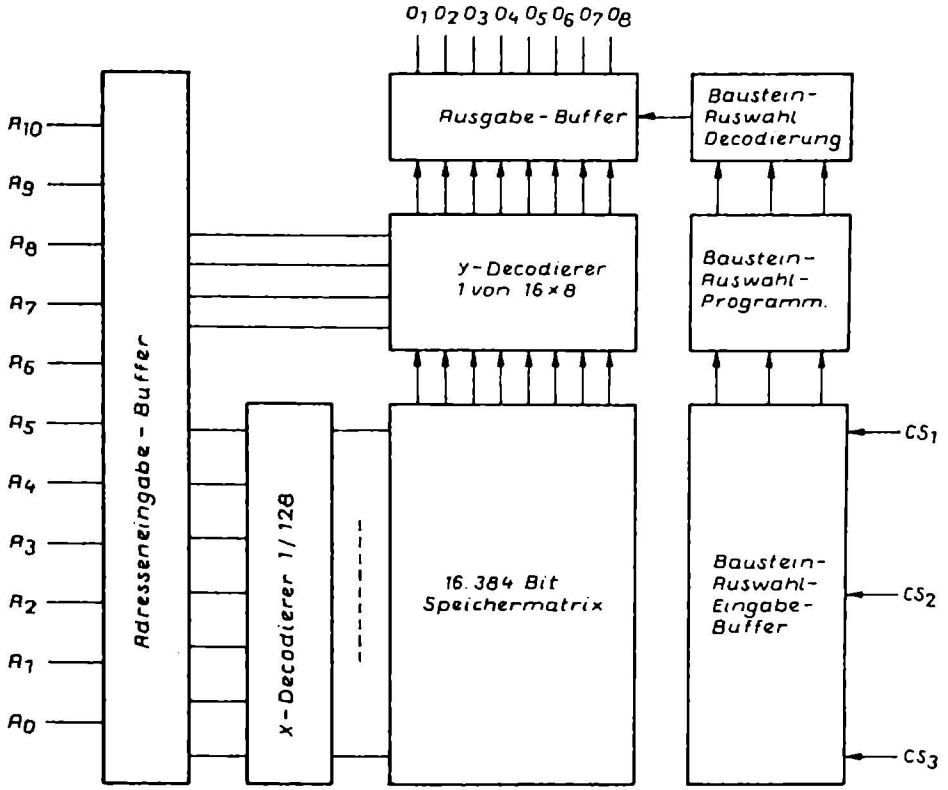


Bild 1.6.12 Aufbau des EPROMs 8716 (16 384-Bits, 2048x8)

In Bild 1.6.12 ist der Aufbau eines 16k-EPROMs gezeigt. Die Organisation für die 16384-Bits beträgt 2048 x 8. Die Eingänge A_0 bis A_{10} dienen zur Adressierung der Zeilen. Ansonsten sind die Anschlüsse mit den vorherigen EPROMs identisch. Die Programmierung kann mit den gleichen Programmspezifikationen der anderen EPROMs durchgeführt werden. Die Programmierung dauert aber ca. 3 Minuten.

Seit kurzer Zeit sind PLAs erhältlich. In Bild 1.6.13 ist die Innenschaltung eines PLAs gezeigt.

An den Eingängen E_1 bis E_{14} liegen die Eingangsinformationen an. Durch die Eingangsnegation wird die Umkehrung der Eingangsvariablen erzeugt. Ein weiteres NICHT-Gatter stellt die Eingangsvariaole wieder her. Durch diese Negationen am Eingang entstehen 28 Leitungen. Aus jedem Eingang wird die direkte Variable und die invertierende Variable erzeugt.
Mit den 96 UND-Gattern erzeugen wir 96 Ausgangsterme. Durch die Programmierung der Eingangsvariablen können wir beliebige konjunktive Verknüpfungen durchführen, da jedes UND-Gatter 14 Eingänge hat. Die Programmierung ist in

Bild 1.6.13 durch ein Kreuz gekennzeichnet. Die konjunktive Verknüpfung erzeugt 96 Ausgangsterme oder Produktterme.
Aus den Produkttermen können die disjunktiven Verknüpfungen durchgeführt werden. Für die disjunktiven Verknüpfungen stehen insgesamt acht ODER-Gatter zur Verfügung. Jedes der acht ODER-Gatter hat 96 Eingänge, die mit den Produkttermen zu verknüpfen sind.
An den Ausgängen der ODER-Gatter sind noch NICHT-Gatter angebracht. Die NICHT-Gatter sind durch die parallel geschaltete Brücke ebenfalls programmierbar. Soll die Ausgangsfunktion eines ODER-Gatters invertiert vorliegen, so ist durch einen Programmierstromimpuls das Sicherungselement zu zerstören.
Das PLA hat folgende Speicherplätze:

Eingangsverknüpfung: 392 x 96 = 57632 Sicherungselemente

Produktterme: 96 x 64 = 6144 Sicherungselemente

Ausgangsnegation: 8 Sicherungselemente

Durch die hohe Anzahl der Sicherungselemente ist ein PLA auch entsprechend teuer. Bei einer logischen Verknüpfung ergibt sich noch eine Berechnung für die minimalisierte Programmierung. Wir sprechen auch hierbei vom Optimum der PLA-Programmierung.
Schalten wir für das PLA einen Decoder, so ergeben sich 2^{14} x 8-Bit-Wörter, d.h. wir haben insgesamt 128k-Speicherplätze zur freien Verfügung.

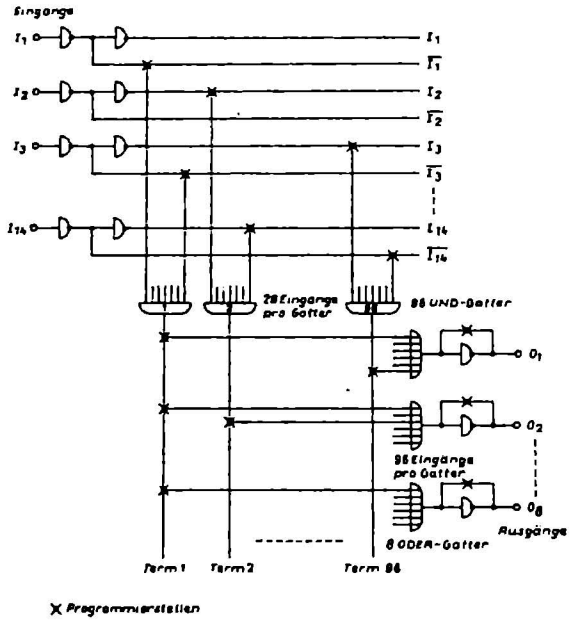


Bild 1.6.13 Aufbau eines PLAs

1.7 Schieberegister

In den letzten Jahren werden in der Elektronik zunehmend Schieberegister zur schnellen Datenzwischenspeicherung verwendet. Dies gilt besonders für die dynamischen MOS-Schieberegister. Die MOS-Register zeichnen sich durch die hohe Packungsdichte aus.

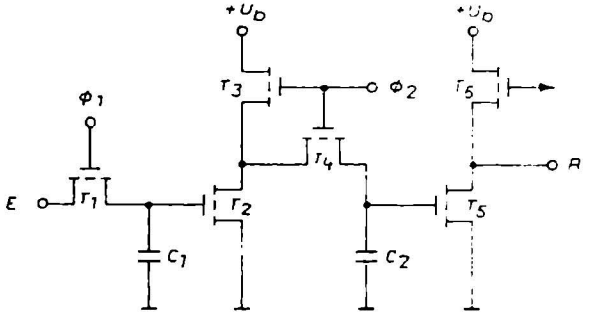


Bild 1.7.1 Aufbau einer dynamischen Speicherzelle in n-Kanal-Technik

In Bild 1.7.1 ist der Aufbau einer dynamische Speicherzelle in n-Kanal-Technik gezeigt. Die Speicherzelle besteht aus sechs Transistoren und zwei Kondensatoren. Die Kondensatoren sind die Gate-Kapazitäten der Transistoren T_2 und T_5.
Für die n-Kanal-Technik gilt, ein H-Signal öffnet den Transistor und ein L-Signal schließt ihn, d.h. einmal wird ein Kanal aufgebaut und einmal abgebaut.
An dem Eingang E soll ein H-Signal anliegen. Der $\varnothing_1$-Takt liegt auf L-Signal. Damit ändert sich nichts, da der Transistor T_1 gesperrt ist. Bei einem H-Signal wird ein Kanal aufgebaut und der Kondensator C_1 kann sich aufladen. Durch die Aufladung des Kondensators erhält der Transistor T_2 ein H-Signal und der Transistor T_2 öffnet sich. An dem Transistor T_4 liegt damit ein L-Signal an. Öffnet sich der Transistor T_3 und T_4, so kann sich der Kondensator nicht aufladen, da er über den Transistor T_2 mit der Masse verbunden ist. Der Kondensator C_2 bleibt ungeladen und an dem Gate von dem Transistor T_5 liegt ein L-Signal an. Der Transistor T_5 ist gesperrt. Erhält der $\varnothing_1$-Takt ein H-Signal, so haben wir an dem Ausgang der Speicherzelle ein H-Signal, da der Transistor T_5 gesperrt ist und der T_6 leitet. An dem Ausgang liegt die Betriebsspannung von +5V an.

Wird nach dem H-Signal ein L eingeschrieben, so entlädt sich der Kondensator C_1 über den Transistor T_1. Durch die Entladung erhält der Transistor T_2 ein L-Signal, d.h. er sperrt. Bei einem H-Signal des Taktes $\varnothing_2$ öffnet sich der Transistor T_3 und T_4 und der Kondensator C_2 lädt sich auf. Wir erhalten ein H-Signal an dem Gate von dem Transistor T_5 und damit liegt der Ausgang A auf L-Signal, da dieser auf Masse geschaltet wird.
Pro Schieberegisterteil ergeben sich zwei Phasenverschiebungen. In der ersten Zelle, dem Master-Teil, wird das Signal um 180° phasenverschoben, ebenso im Slave-Teil. Bei jedem Taktsignal wird die Eingangsinformation um eine Stelle weitergeschoben. Daher müssen die beiden Taktphasen $\varnothing_1$ und $\varnothing_2$ um 180° phasenverschoben sein. Damit ist eine sichere Weiterschiebung der Daten gewährleistet.

Ein wesentliches Problem bei den dynamischen Schieberegistern ist das Auffrischen der Gate-Kapazitäten (Kondensatoren) in den einzelnen Speicherzellen. Das Auffrischen wird durch die beiden Taktphasen vorgenommen. Bei einer Informationsverschiebung um eine Stelle wird gleichzeitig das Signal aufgefrischt. Die Kapazität der Gate-Kondensatoren liegt in der Größenordnung von 0,1pF bei einer Umgebungstemperatur von 25°C. Ist der Gate-Kondensator aufgeladen, fließt ein Reststrom von ca. 0,1pA. Dieser Reststrom ist temperaturabhängig. Die Gate-Kapazitäten müssen daher immer auf- oder entladen werden.
Der Auffrisch-Zyklus soll nach ca. 1ms erfolgen, d.h. die Information wird durch diesen Zyklus um eine Stelle pro 1ms weitergeschoben. Verwendet man ein Schieberegister mit 1000 Speicherzellen, so ist die Eingangsinformation nach einer Sekunde nach dem Einschreibzeitpunkt am Ausgang wieder vorhanden. Damit kann durch die Taktfrequenz der beiden Systeme die Schiebegeschwindigkeit bestimmt werden.
Die Taktfrequenz für die beiden Phasentakte $\emptyset_1$ und $\overline{\emptyset}_2$ kann einfach erzeugt werden. Ein T-Flip-Flop hat zwei Ausgänge, den Q- und $\overline{Q}$-Ausgang. Beide Ausgänge sind um 180° phasenverschoben. Für den Betrieb des dynamischen Schieberegisters reicht diese Phasenverschiebung aus.

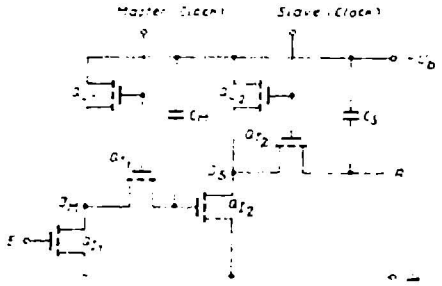


Bild 1.7.2.1 Innenschaltung einer p-Kanal-MOS-Schieberegister-Zelle

In der Technik verwendet man meistens p-Kanal-MOS-Schieberegister In Bild 1.7.2 ist die Innenschaltung gezeigt. Liegt an dem Eingang E ein H-Signal an, so kann sich die Gate-Kapazität C_M über die beiden Transistoren Q_{T1} und Q_{L1} entladen. Der Spannungspunkt J_M hat nach der Entladung ca. -5V. Die Entladung kann aber nur stattfinden, wenn der Master Clock $\emptyset_M$ ($\emptyset_1$) auf L-Signal geht (+5V nach -12V). Beim Slave-Clock $\emptyset_S$ ($\emptyset_2$) ist der Eingangs-Transistor Q_{T2} der Slave-Zeile gesperrt. Der Längstransistor Q_{L2} wird leitend und die Gate-Kapazität C_S kann sich über den Transistor Q_{T2} aufladen. Am Ausgang erscheint ein H-Signal, das mit dem Eingangssignal phasenidentisch ist. Die Master- und die Slave-Speicherzelle invertieren jedes Eingangssignal um 180°. Durch die doppelte Phasenverschiebung ergibt sich am Ausgang ein phasenidentisches Signal.
Liegt an dem Eingang E ein L-Signal (-5V), ist der Transistor Q_{T1} gesperrt. Die Gate-Kapazität C_M kann sich über die beiden Transistoren Q_{L1} und Q_{T1} aufladen, wenn ein Master-Clock $\emptyset_M$ anliegt. Die Ladung des Kondensators öffnet den Transistor Q_{T2} der Slave-Zelle. Die Gate-Kapazität CS kann sich beim Slave-Clock $\emptyset_S$ über den Transistor Q_{T2} und Q_{T2} entladen. Am Ausgang A der Slave-Speicherzelle erscheint ein L-Signal, das wieder mit dem Eingangssignal identisch ist.

In Bild 1.7.2 sind die einzelnen Spannungen in der dynamischen MOS-Speicherzelle gezeigt. Aus dem Spannungsdiagramm ersehen wir, daß bei zu langen Taktintervallen die Gate-Kapazitäten C_M und C_S entladen werden können und damit geht die gespeicherte Information verloren. Die Größe der Gate-Kapazitäten liegt in der Größenordnung von 0,1pF. Bei einer Umgebungstemperatur von 25°C fließt ein Reststrom von 0,1pA, d.h. die Speicherzellen müssen nach spätestens 1ms aufgefrischt werden.

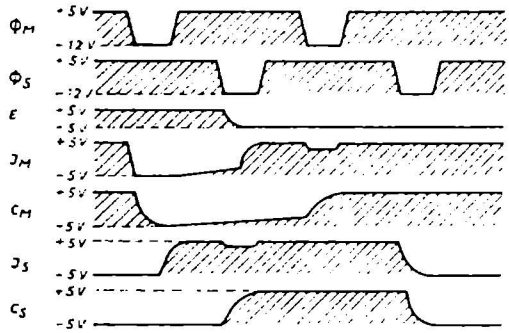


Bild 1.7.2.2 Spannungsdiagramm für eine p-Kanal-MOS-Schieberegister-Zelle

Die Auffrischung der Speicherzellen von Bild 1.7.2 kann mit der Treiberschaltung von Bild 1.7.3 durchgeführt werden. An dem Eingang T der Treiberschaltung liegt der Phasentakt an, der von einem TTL-Baustein erzeugt wird. Das Eingangs-TTL-Gatter hat an dem Ausgang einen offenen Collektor und daher ist ein Arbeitswiderstand erforderlich. Ist der Ausgangs-Transistor des TTL-Gatters gesperrt, dann liegt an dem pnp-Transistor eine Spannung von +5V an. Damit ist der Transistor gesperrt. Über die Widerstände liegen an dem Ø-Ausgang -12V an. Leitet dage-

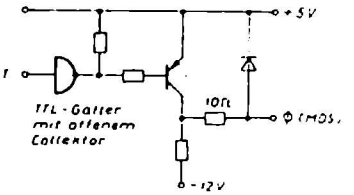


Bild 1.7.3 Treiberschaltung für ein p-Kanal-Schieberegister

gen der Ausgangs-Transistor das Gatter, schaltet der pnp-Transistor durch. An dem Ausgang der Treiberschaltung liegt damit eine +5V-Spannung an.

Für den Betrieb der Speicherzelle von Bild 1.7.2 sind zwei Taktsysteme erforderlich. Pro Taktsystem können wir die Treiberschaltung von Bild 1.7.3 verwenden. Als Ansteuerung für die beiden Treiberschaltungen dient uns ein T-Flip-Flop. Der Treiber für den Master-Takt liegt z.B. an dem Q-Ausgang und der Slave-Takt an

dem $\overline{Q}$-Ausgang. Damit ist eine Phasenverschiebung von 180° garantiert. Häufig schaltet man zwischen dem T-Flip-Flop und den beiden Treiberschaltungen Monoflops ein.

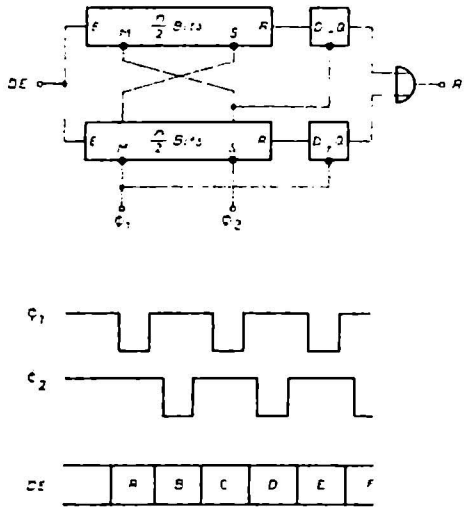


Bild 1.7.4 Speicherprinzip eines dynamischen Schieberegisters

In Bild 1.7.4 ist das Speicherprinzip bei dynamischen Schieberegistern gezeigt. An den beiden DE-Eingängen liegt die Eingangsinformation an. Mittels den beiden Taktsystemen wird die Eingangsinformation eingeschoben und weitergeschoben. Die beiden Schieberegister-Systeme sind durch die kreuzweise Verbindung ebenfalls um 180° phasenverschoben. Damit kann das eine Schieberegister eine Information aufnehmen und das andere schiebt zwischenzeitlich die Information weiter. Beim nächsten Taktschritt ist dies genau umgekehrt. An den Ausgängen der beiden Schieberegister befinden sich zwei D-Flip-Flops. Die beiden D-Flip-Flops speichern die eingeschriebene Information kurz zwischen und geben sie so auf den Ausgang. Durch das ODER-Gatter erhalten wir eine Impulsverlängerung in Verbindung mit den beiden Flip-Flops.
Die eingeschriebenen Daten von Bild 1.7.4 gehen nach der Speicherung verloren, da anschließend keine Zwischenspeicherung mehr vorhanden ist. Diesen Nachteil hat die Schaltung von Bild 1.7.5 nicht. Wir haben hier ein rezirkulierendes Schieberegister. Durch die Steuerung mit dem Eingang S des Multiplexers können wir bestimmen, ob die Daten umlaufen (rezirkulieren) oder ob Daten eingeschrieben werden sollen.
Mit einem L-Signal an dem S-Eingang des Multiplexers schaltet der Baustein so durch, daß die Ausgangsdaten des Schieberegisters direkt mit den Eingängen wieder verbunden sind. Die Daten laufen hierbei um, d.h. die Ausgangsdaten werden am Eingang wieder neu eingeschrieben. Die Umlaufgeschwindigkeit ist von der Taktfrequenz der beiden Phasentakte $\varnothing_1$ und $\varnothing_2$ abhängig. Die umlaufenden Daten können direkt an den DA-Ausgängen zerstörungsfrei abgenommen werden.

Neue Daten werden in den Datenumlauf eingeschrieben, wenn wir den S-Eingang des Multiplexers auf H-Signal legen. Die Eingangsdaten DE werden sofort eingeschrieben, d.h. die Einschreibegeschwindigkeit hängt von der Taktfrequenz der Phasentakte ab. Die Daten des Schieberegister-Ausganges gehen dabei verloren, da keine Verbindung mehr zu den Daten-Eingängen besteht.

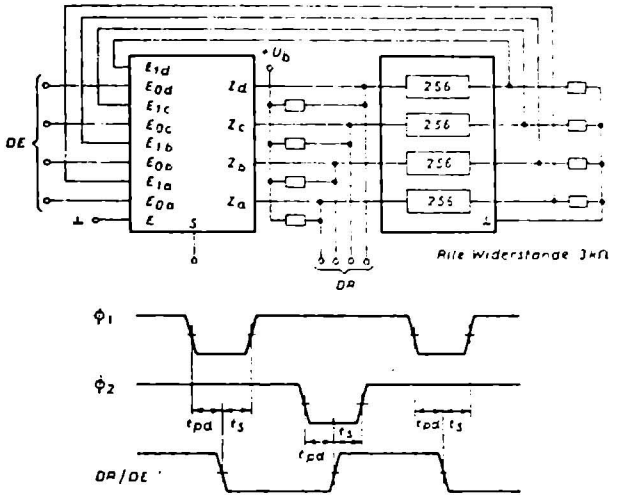


Bild 1.7.5 Rezirkulierendes Schieberegister

Die neu eingeschriebenen Daten stehen sofort an dem DA-Eingang zur Verfügung, danach aber erst wieder nach 256 Impulsen. Die Schaltung von Bild 1.7.5 hat den Vorteil, daß wir den Taktbetrieb umschalten können. Das Umschalten muß aber prellfrei erfolgen, da es sonst zu Datenfehlern kommen kann.
Im Ruhebetrieb läuft das Schieberegister mit einer Taktfrequenz von 2kHz, d.h. durch die Frequenzteilung des T-Flip-Flops operiert das Schieberegister mit einem Phasentakt von 1 kHz. Bei Einschreiben der Informationen können wir durch einen elektronischen Umschalter die Eingangsfrequenz des T-Flip-Flops auf 2MHz erhöhen, d.h. durch die Frequenzteilung läuft nun das Schieberegister mit einem Phasentakt von 1MHz um. Dadurch können die Daten wesentlich schneller eingeschrieben werden.

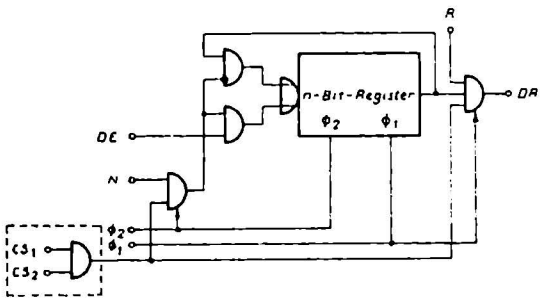


Bild 1.7.6 Dynamische Schieberegister mit interner Logik für den rezirkulierenden Betrieb

In Bild 1.7.6 ist ein dynamisches Schieberegister mit interner Logik für den rezirkulierenden Betrieb gezeigt. Liegt an einem der beiden CS-Eingänge ein L-Signal an, ist die UND-Bedingung nicht erfüllt und die Daten laufen um. Das Schieberegister rezirkuliert. Zu diesem Zeitpunkt ist der Schreibeingang W (write) und der Daten-Eingang DE gesperrt. Die umlaufenden Daten können bei diesem Schieberegister nicht ausgelesen werden.
Die beiden CS-Eingänge liegen an einem H-Signal. Die UND-Bedingung ist erfüllt und die umlaufenden Daten können über das Ausgangs-UND-Gatter ausgelesen werden, wenn der Lese-Eingang R(read) auf H-Signal liegt. Liegt der Schreib-Eingang W an L-Signal, können keine Daten über den DE-Eingang eingeschrieben werden. Das Register läuft um, aber die Daten können am DA-Ausgang ausgelesen werden und zwar störungsfrei .
Legen wir dagegen ein H-Signal an den Eingang W, ist das Umlauf-UND-Gatter gesperrt und die Informationen des Eingangs DE können eingeschrieben werden. Die Daten am Eingang werden eingeschrieben, wenn der Takt $\emptyset_2$ an einem H-Signal liegt. Zu den anderen Zeiten ist die Daten-Aufnahme blockiert. Die Daten-Ausgabe erfolgt nur, wenn der Takt $\emptyset_1$ ein H-Signal hat.

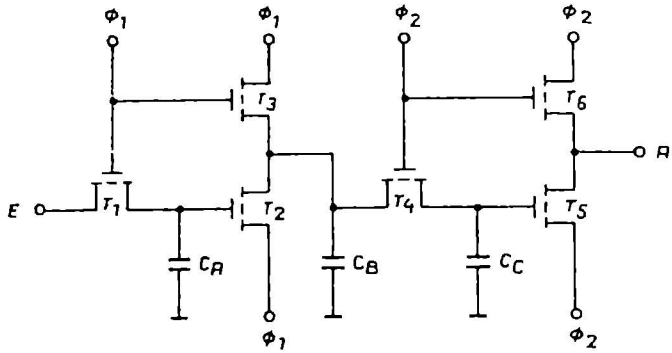


Bild 1.7.7 Schieberegister-Zelle in n-Kanal-Technik

In Bild 1.7.7 ist ein Schieberegister in n-Kanal-Technik gezeigt. Für dieses Schieberegister ist eine wesentlich einfachere Treiber-Schaltung erforderlich. An dem Eingang E liegt ein H-Signal an. Das H-Signal wird in die Speicherzelle übernommen, wenn der Takt $\emptyset_1$ ein H-Signal hat. Damit wird der Transistor T_1 leitend und das Eingangssignal kann den Kondensator C_A aufladen. Nach 0,7 τ beginnt der Transistor T_2 zu leiten. Über den Transistor T_2 liegt das H-Signal des Phasentaktes an dem Kondensator C_B an. Damit kann sich der Kondensator C_B aufladen.
Beim Phasentakt $\emptyset_2$ wird der Transistor T_4 leitend und der Kondensator C_C lädt sich auf. Damit hat das Gate von dem Transistor T_5 ein H-Signal und das H-Signal des Phasentaktes $\emptyset_2$ kann den Kondensator C_D aufladen. Der wesentliche Unterschied bei dem Schieberegister von Bild 1.7.7 ist die Tatsache, daß hierbei keine Betriebsspannung erforderlich ist. Die Ladung der drei Kondensatoren wird mit-

tels den zwei Phasentakten erzeugt. Durch die Ladungsspeicherung der drei Kondensatoren tritt keine Phasenverschiebung auf.
Liegt an dem Eingang E ein L-Signal an, wird bei einem H-Signal des Phasentaktes $\emptyset_1$ der Kondensator C_A entladen. Damit wird der Transistor T_2 gesperrt. Hat der Phasentakt $\emptyset_1$ ein L-Signal, so kann sich der Kondensator C_B über den Transistor etwas aufladen. Die Ladung des Kondensators C_B verringert sich sofort um die Hälfte, wenn der Phasentakt $\emptyset_2$ ein H-Signal bekommt. Damit liegt an dem Transistor T_5 ein L-Signal an. Der Transistor ist gesperrt und der Kondensator C_D kann sich wieder etwas aufladen.
Die Ladung durch die beiden Transistoren T_3 und T_6 ist sehr gering. Die Entladung der internen Kondensatoren hängt immer von der Parallelschaltung der beiden Kondensatoren C_B mit C_C ab. Beim Diffundieren wird hier bereits auf einen relativen hohen Reststrom hingearbeitet. Daher werden diese Schieberegister-Typen kaum in der Elektronik eingesetzt.

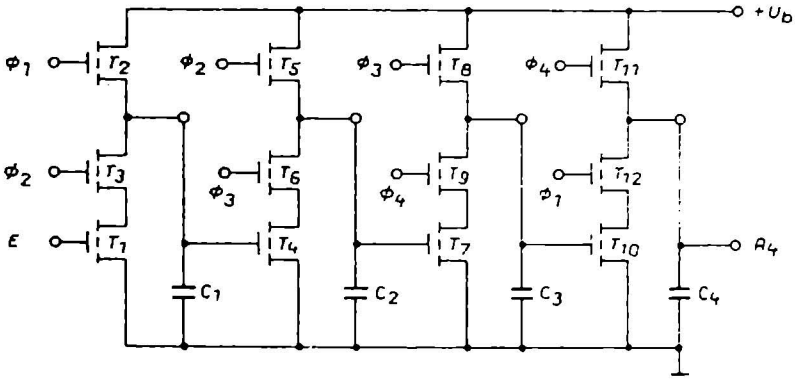


Bild 1.7.8 Aufbau einer Zelle von einem Vierphasen-Schieberegister

Wesentlich interessanter ist das Vierphasen-Schieberegister von Bild 1.7.8. Jede Speicherzelle besteht hier aus vier MOS-Kondensatoren. Der untere Eingang ist der Daten-Eingang. Dieser ist direkt mit dem vorherigen Ausgang verbunden. Anhand des Bildes kann bereits die Taktverschiebung bei den Speicherzellen gezeigt werden.
An dem Eingang E soll ein H-Signal anliegen. Dadurch wird der Transistor T_1 leitend. Beim H-Signal des Taktes $\emptyset_1$ lädt sich der Kondesator C_1 mit H-Signal auf. Der Kondensator wird aber entladen, wenn der Transistor T_3 ein H-Signal durch den Phasentakt $\emptyset_2$ erhält. Gleichzeitig kann sich der Kondensator C_2 aufladen, da der Transistor T_5 ebenfalls an dem Phasentakt $\emptyset_2$ liegt. Der Kondensator C_1 entlädt sich über den Transistor T_1 und T_3. Damit hat die zweite Teilstufe der Speicherzelle an dem Eingang E_1 ein L-Signal. Dieses L-Signal sperrt den Transistor T_4. Erhält der Gate-Anschluß des Transistors T_6 ein H-Signal durch den Phasentakt $\emptyset_3$, so kann sie der Kondensator C_2 nicht entladen, da der Transistor T_4 gesperrt ist.

Der Eingang E_2 hat durch die Kondensatorladung von C_2 an dem Gate-Anschluß des Transistors T_7 ein H-Signal. Damit ist der Transistor T_7 leitend. Mit dem Phasentakt $\emptyset_3$ hat sich bereits der Kondensator C_3 aufgeladen. Der Kondensator C_3 entlädt sich aber mit dem Phasentakt $\emptyset_4$ wieder über den Transistor T_7 und Tg. Der Kondensator C_3 hat ein L-Signal. Dieses L-Signal liegt an dem Eingang E_3, der letzten Stufe der Speicherzelle an. Der Transistor T_{10} wird durch das L-Signal des Einganges E_3 gesperrt. Bereits zu dem Takt $\emptyset_4$ kann sich der Kondensator C_4 aufladen. Der Kondensator kann sich beim Phasentakt $\emptyset_1$ nicht entladen, da der Transistor T_{10} gesperrt ist. Die Speicherzelle von Bild 1.7.8 benötigt vier Phasensysteme zum Weitertransport der Daten. Als Phasentakt-Generator kann in diesem Fall ein Schieberegister verwendet werden. Ein 4-Bit-Schieberegister wird mit einem H-Signal geladen. Dieses H-Signal wird pro Taktimpuls um eine Stelle weitergeschoben. Pro Schieberegister-Ausgang erhalten wir dadurch einen Phasentakt. Bevor die statischen Schieberegister behandelt werden, soll der Aufbau und die Wirkungsweise von dynamischen CMOS-Schieberegistern gezeigt werden. In Bild 1.7.9 ist das Prinzip einer dynamischen CMOS-Speicherzelle gezeigt. Die Speicherung der Eingangsdaten erfolgt durch die beiden Kondensatoren C_1 und C_2.
Die beiden Schalter S_1 und S_2 sind miteinander verbunden. Um 180^0 phasenverschoben arbeitet dagegen der Schalter S_3, d.h. sind die beiden Schalter S_1 und S_2 offen, so ist S_3 geschlossen. und umgekehrt.

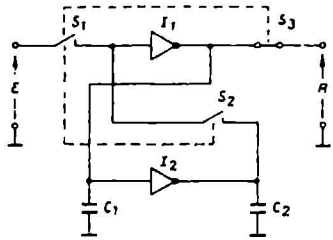


Bild 1.7.9 Prinzip einer dynamischen CMOS-Speicherzelle

Liegt an dem Eingang E ein H-Signal an, so liegt die Eingangsinformation über den Schalter S_1 an dem CMOS-Inverter. Der Inverter invertiert das Signal und der Kondensator C_1 kann sich nicht aufladen. Das L-Signal des Ausgangs von I_1 wird durch den zweiten Inverter invertiert und der Kondensator C_2 kann sich aufladen. Öffnen wir nun die Schalter S_1 und S_2, so liegt an dem Inverter I_1 ein H-Signal. Das H-Signal wird durch den Inverter I_1 umgekehrt und an dem Ausgang A liegt ein L-Signal, da der Schalter S_3 geschlossen ist.
Wird in die Speicherzelle ein L-Signal geschrieben, so erhält der Kondensator C_1 ein H-Signal, d.h. er kann sich über den Inverter I_1 aufladen. Damit wird aber der Kondensator C_2 entladen, da durch den Inverter I_2 das H-Signal invertiert wird. Schließt der Schalter S_3, so liegt an dem Ausgang A ein H-Signal.
Beim CMOS-Schieberegister tritt zwischen dem Eingang und dem Ausgang eine Phasendrehung von 180^0 auf. Da die Information immer mit zwei Taktphasen weitergeschoben wird, ist theoretisch nach der Speicherzelle nach Bild 1.7.9 noch ein Inverter erforderlich. Diesen Inverter können wir aber vergessen, wenn es sich

um geradzahlige Schieberegister handelt, d.h. 8, 100, 256, 1024, oder 2048-Bits lang sind. Pro Zelle wird die Information negiert. Dadurch erhalten wir am Ausgang wieder ein phasenidentisches Signal zum Eingang.

In Bild 1.7.10 ist der interne Aufbau eines dynamischen CMOS-Schieberegisters gezeigt. Die beiden Inverter werden durch die Zusammenschaltung der beiden p-Transistoren mit den n-Transistoren hergestellt. Liegt an dem gemeinsamen Gate eines Inverters ein H-Signal an, so ist der p-Transistor gesperrt, aber der n-Transistor leitet. Wechselt das Signal an dem gemeinsamen Gate von H nach L, so ist der obere Transistor (p-Kanal) leitend, aber der untere n-Kanal-Typ gesperrt. Daher ergibt sich in der CMOS-Technik durch das Zusammenschalten eines p- und n-Kanal-Transistors eine Phasenverschiebung von 180^{o} zwischen Ausgang und Eingang.
An dem Eingangstransistor (p-Kanal) liegt ein H-Signal an. Der Transistor ist leitend, wenn der Phasentakt Ø ein L-Signal hat. An dem Eingangsinverter I_1 liegt an dem gemeinsamen Gate ein H-Signal und der Ausgang des Inverters I_2 hat ein L-Signal. Der Kondensator C_1 kann sich nicht aufladen. Das L-Signal wird invertiert und der Kondensator C_2 lädt sich auf.
Der Schalter S_1 ist ein CMOS-Transmissions-Gatter. Hierbei liegt ein p- und ein n-Kanal-Transistor parallel und wir steuern sie mit einem gemeinsamen Takt an, der aber um 180^{o} phasenverschoben ist. Geben wir auf den Ø-Takt ein L-Signal, so hat der $\overline{Ø}$-Takt ein H-Signal. Der p-Transistor befindet sich ebenso im leitenden Zustand, wie der n-Kanal-Transistor. Die Ladung des Kondensators C_2 liegt an dem Eingang des Inverters I_1.
Hat der Takt Ø ein L-Signal, so ist der Schalter S_3 leitend. Die beiden anderen Schalter sind zu dieser Zeit gesperrt. Die gespeicherte Information des Kondensators C_2 liegt über dem Inverter I_1 an dem Ausgang an. Bei der Einschreibung eines L-Signales wird der Kondensator C_2 entladen. Damit ist in der Speicherzelle ein L-Signal gespeichert.

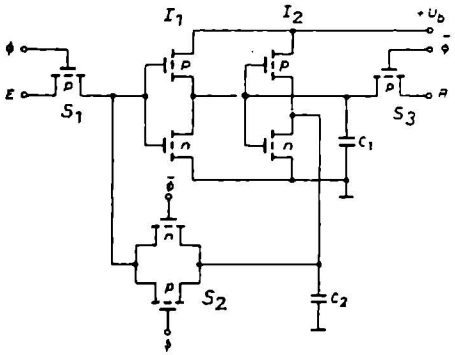


Bild 1.7.10 Interner Aufbau einer dynamischen CMOS-Speicherzelle

In Bild 1.7.11 ist eine statische MOS-Schieberegister-Speicherzelle gezeigt. Für den Betrieb einer statischen MOS-Speicherzelle sind drei Taktsysteme erforderlich. Mit dem ersten Taktsystem $\emptyset_1$ wird die Eingangsinformation in das Schieberegister eingeschrieben. Die beiden anderen Phasentakte $\emptyset_2$ und $\emptyset_3$ sind für die Auffrischung der beiden Gate-Kapazitäten C_A und C_B erforderlich.

Im wesentlichen stellt die Speicherzelle von Bild 1.7.11 eine bistabile Kippstufe dar. Die beiden Transistoren T_2 und T_3 sind mit den Schalttransistoren einer normalen Kippstufe identisch. Die beiden Transistoren T_6 und T_7 sind als Arbeitswiderstände geschaltet, d.h. als Strombegrenzungswiderstände.

Geschaltet wird die Speicherzelle von Bild 1.7.11 durch ein Eingangssignal an E. Der Eingang E soll ein H-Signal haben. Hat der n-Kanal-Transistor T_1 an dem $\emptyset_1$-Takt ein H-Signal, so liegt dieses H-Signal an dem Gate des Transistors und an der Gate-Kapazität C_A. Der Kondensator C_A kann sich aufladen und damit schaltet der Transistor T_2 durch. Vernachlässigen wir den Transistor T_4, so bedeutet das für den Transistor T_3, daß er auf Masse liegt. Der Kondensator C_B (Gate-Kapazität des T_3) kann sich nicht aufladen und der Ausgang A der bistabilen Kippstufe liegt auf H-Signal.

Der Kondensator C_A ist geladen und der Kondensator C_B entladen. Mittels des Phasentaktes von $\emptyset_2$ und $\emptyset_3$ werden die beiden Kondensatorladungen laufend auf- oder entladen. Der Kondensator C_A kann sich über den Transistor T_5 laufend aufladen, da der Ausgang A ein H-Signal aufweist. Dagegen wird der Kondensator C_B laufend über den Transistor T_4 entladen, da der Transistor T_2 durchgeschaltet hat und damit auf Masse liegt. Die Ladung der beiden Kondensatoren wird laufend erneuert.

An dem Eingang E soll ein L-Signal anliegen. Mit einem H-Signal an dem Takt $\emptyset_1$ wird der Kondensator C_A entladen und das Gate des Transistors T_2 sperrt.

Dadurch liegt ein H-Signal an dem Gate von dem Transistor T_3, wenn wir momentan die Funktion des Transistors T_4 vernachlässigen. Der Kondensator C_B lädt sich auf und der Transistor T_3 ist leitend. An dem Ausgang A haben wir daher ein L-Signal.

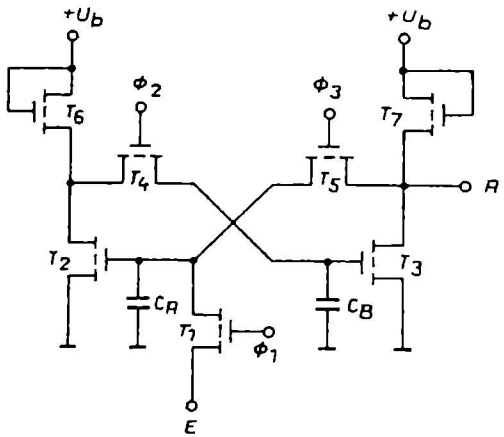


Bild 1.7.11 Aufbau einer statischen MOS-Schieberegister-Speicherzelle

Die Auffrischung der beiden Kondensator-Ladungen wird wieder über die beiden Transistoren T_4 und T_5 vorgenommen. Mit einem H-Signal schalten die Transistoren durch und der Kondensator C_A liegt an einem L-Signal und der Kondensator C_B an einem H-Signal. Die beiden Ladungen werden laufend erneuert. Die beiden Taktsysteme werden von dem Schieberegister intern erzeugt. Wegen der günstigen Leistungsaufnahme arbeiten die beiden Taktsysteme für $\emptyset_2$ und $\emptyset_3$ um 180^o phasenverschoben. Hat der Takt $\emptyset_2$ ein H-Signal, können sich alle Transistoren T_4 gleichzeitig öffnen. Die Kondensatoren C_B werden auf- oder entladen. Mit dem Takt $\emptyset_3$ laden oder entladen sich die Kondensatoren C_B, da die Transistoren T_5 ein H-Signal erhalten. Die Zellenauffrischung kann intern erzeugt werden. Hierzu ist ein astabiler Multivibrator eingebaut, der auf einer Frequenz von 5MHz schwingt. Dieser steuert ein T-Flip-Flop an. An dem Q-Ausgang des T-Flip-Flops ist der $\emptyset_2$ angeschlossen und an dem $\overline{Q}$-Ausgang der $\emptyset_3$-Takt.

Das Weiterschieben der Information geht nur durch den $\emptyset_1$-Takt. Der $\emptyset_1$-Takt arbeitet völlig unabhängig von dem internen Auffrischungs-Takt. Alle $\emptyset_1$-Gate-Anschlüsse liegen parallel. Bei jedem H-Signal wird eine Information eingeschrieben und die gespeicherten Informationen weitergeschoben. Die Einschreibegeschwindigkeit beträgt maximal 1MHz.

Soll die Einschreibegeschwindigkeit erhöht werden, so ist der interne Oszillator in seiner Frequenz zu erhöhen. Diese Frequenzerhöhung kann aber nur von dem Halbleiter-Hersteller durchgeführt werden.

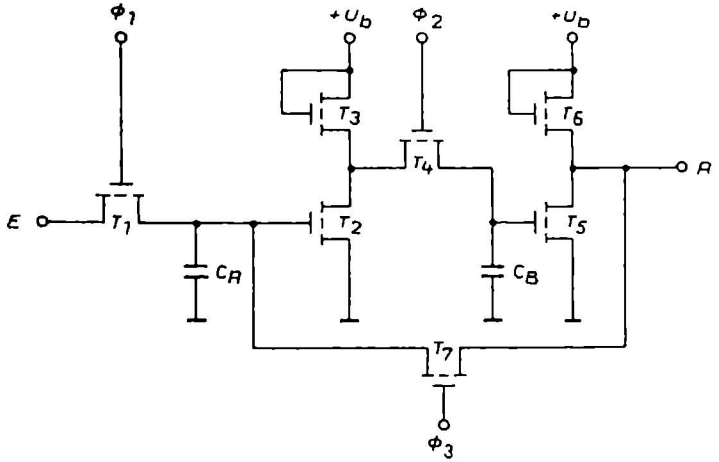


Bild 1.7.12 Aufbau einer statischen MOS-Schieberegister-Speicherzelle

Eine andere MOS-Speicherzelle für den statischen Betrieb ist in Bild 1.7.12 gezeigt. Liegt an dem Eingang E ein H-Signal an so kann dieses Signal eingeschrieben werden, wenn an dem $\emptyset_1$-Takt ein H-Signal anliegt. Der Kondensator C_A kann sich aufladen. Dadurch liegt das Gate des Transistors T_2 an einem H-Signal. Der Transistor T_2 ist daher durchgeschaltet. Mit dem Phasentakt $\emptyset_2$ entlädt sich der Kondensator C_B. Damit liegt an dem Gate vom Transistor T_5 ein L-Signal an und der Transistor ist gesperrt. Der Ausgang der Speicherzelle hat damit ein H-Signal. Dieses H-Signal liegt an der nächsten Eingangsstufe an.

Im statischen Betrieb erfolgt nun ein H-Signal auf den $\emptyset_3$-Phasentakt. Damit

wird der Transistor T_7 leitend und der Kondensator C_A kann sich aufladen. Diese Aufladung ist aber sehr gering, da der Kondensator bereits aufgeladen ist. Der Kondensator wird nur aufgefrischt.
Durch die beiden Taktsysteme $\varnothing_2$ und $\varnothing_3$ werden die Ladungen der beiden Kondensatoren immer nur aufgefrischt. Das Weiterschieben der Informationen wird durch den $\varnothing_1$ vorgenommen.

2. Aufbau eines Mikroprozessor-Systems mit dem 8080

In dem Bild 2.1 ist der Aufbau eines Mikroprozessor-Systems mit dem 8080 gezeigt. Das System besteht aus sechs Bausteinen:

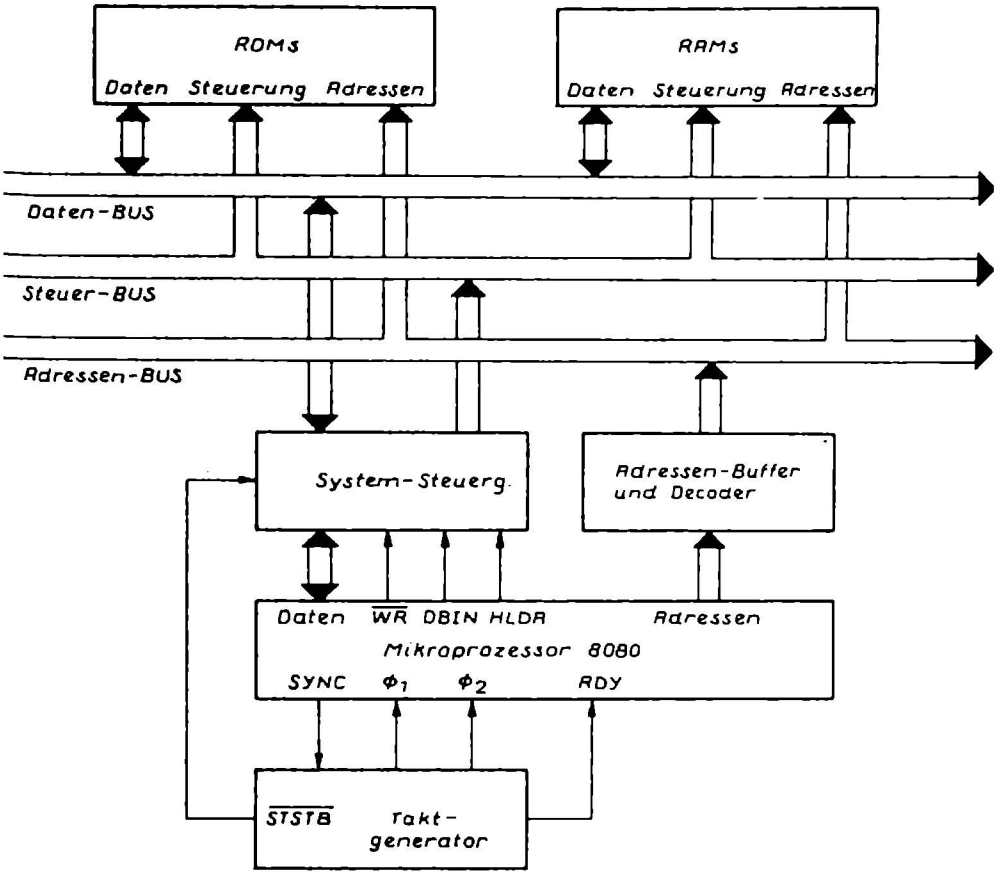


Bild 2.1 Aufbau des Mikroprozessor-Systems 8080

1) Der Mikroprozessor 8080 bildet den Mittelpunkt der Schaltung. Der 8080 benötigt für den Betrieb einen Zweiphasen-Takt, den er von dem Taktgenerator bezieht. Die Daten bezieht der Mikroprozessor aus dem Daten-BUS. Ebenso gibt er seine Daten in den Daten-BUS wieder zurück. Die Adressen erzeugt der interne Befehlszähler im 8080. Die Adressen werden über einen Buffer auf den Adressen-BUS geschaltet.

2) Der Taktgenerator ist ein Oszillator. Bei dem Taktgenerator soll der Baustein 8224 unbedingt verwendet werden. Der 8224 liefert dem 8080 seine beiden Phasentakte. Die Phasentakte werden durch den SYNC-Eingang synchronisiert.

3) Durch die System-Steuerung kann der Datenfluß von dem Mikroprozessor abgeschaltet werden. Der Mikroprozessor ist damit in der Lage, unabhängig von

dem Daten-BUS zu arbeiten. Die System-Steuerung ist ein Zweiwege-System, d.h. über das Baustein-System können Daten von oben nach unten, also in den Mikroprozessor oder von unten nach oben, und damit aus dem Mikroprozessor gelangen.

4) Der Adressen-Buffer ist nur dann erforderlich, wenn mehrere Bausteine an dem Adressen-BUS angeschlossen sind.

5) Das ROM wird als Programmspeicher benötigt. Hierbei haben wir die Möglichkeit, entweder Festwertspeicher oder EPROMs bei der Entwicklung einzusetzen. Die Daten-Ausgabe des ROMs erfolgt direkt auf den Daten-BUS. Die Adressierung des ROMs erfolgt durch den Adressen-BUS. Ist die Adressierung abgeschlossen, so erscheint auf dem Steuer-BUS ein Freigabe-Impuls und die Daten des ROMs werden auf den Daten-BUS geschaltet.

6) Mit dem RAM können wir die Programmdaten, Übersichtstabellen, Zwischenwerte und die Adresse des Stack-Pointers (Kellerspeicher) ablegen. Da für das Einschreiben oder Auslesen zwei verschiedene Signale erforderlich sind, ist die Steuer-Leitung etwas umfangreicher.

2.1 Der Mikroprozessor 8080

Über den Mikroprozessor 8080 wurde bereits im Mikroprozessor-Buch Teil 1 geschrieben.

Der Typ 8080 ist ein Ein-Chip-Mikroprozessor mit einer 8-Bit-parallel-Struktur für allgemeine Anwendungsbeispiele. Die Daten und Zustandsinformationen werden auf einem 8-Bit-Daten-Bus (D_0 bis D_7) und die Adressen für die Speichereinheiten und Peripheriegeräte über einen separaten 16-Bit-Adressen-Bus (A_0 bis A_{15}) übertragen.

Der Typ 8080 hat sechs Zeitsignal- und vier Steuer-Ausgänge:

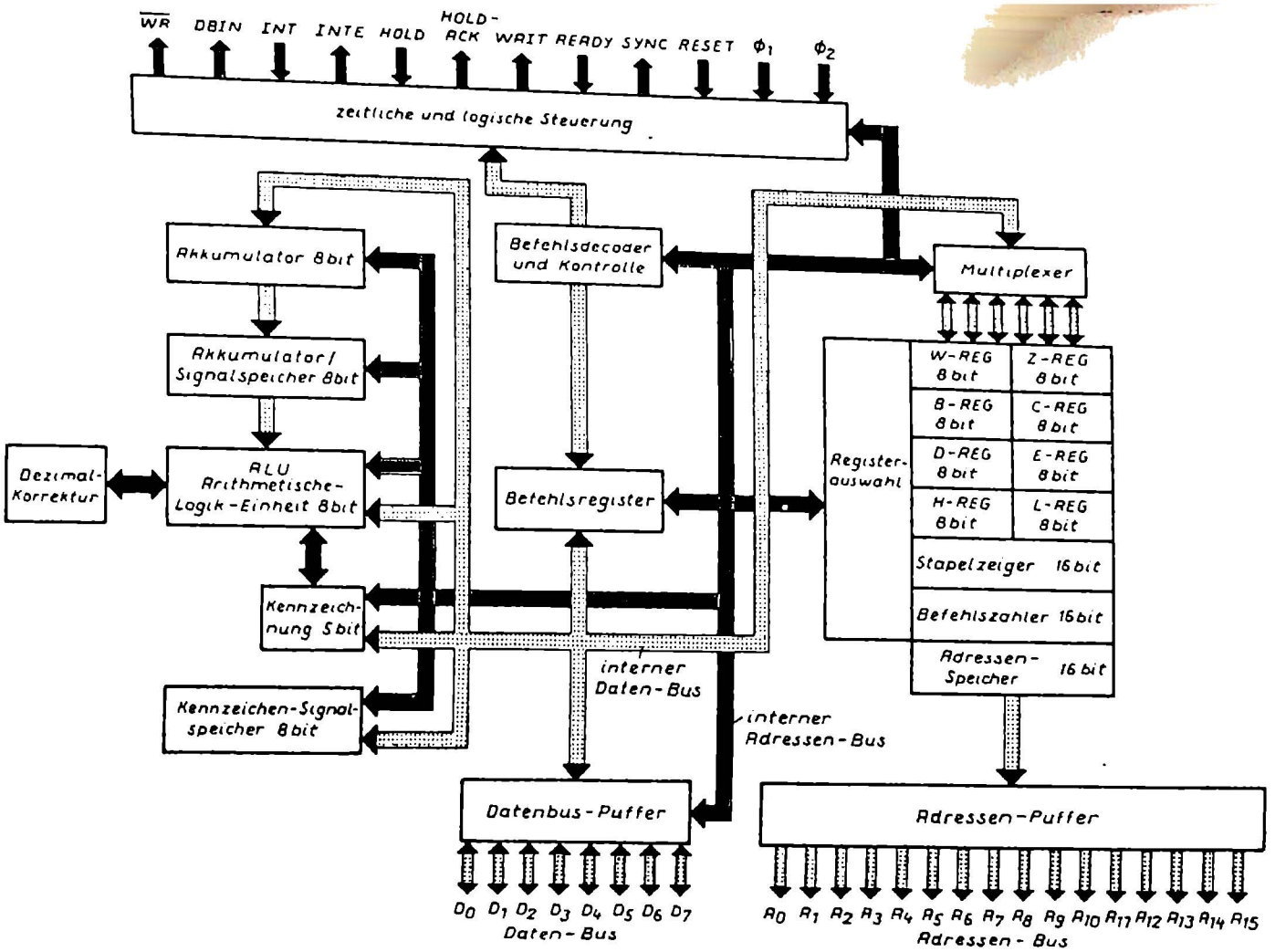
SYNC	=	Ausgang zum Synchronisieren von externen Einrichtungen
DBIN	=	Daten-Bus-Steuerung für externe Einrichtungen
WAIT	=	Ausgang für die Warte-Steuerung
WR	=	Schreib-Ausgang
HLDA	=	Ausgang für die Halt-Steuerung
INTE	=	Ausgang für die Unterbrechungs-Steuerung

Vier Steuer-Eingänge:

READY	=	Eingang für die interne Warte-Steuerung
HOLD	=	Eingang für die interne Halt-Steuerung
INT	=	Eingang für die interne Unterbrechungs-Steuerung
RESET	=	Zum Löschen und Zurückstellen des Mikroprozessors

Der Typ 8080 hat noch vier Betriebsspannungs-Eingänge (+ 12V, + 5V, 0V und - 5V) und zwei Takt-Eingänge ($\emptyset_1$ und $\emptyset_2$).

In Bild 2.1.1 ist das Blockschaltbild des 8080 gezeigt. Dieses Blockschaltbild zeigt vier große Funktionsblöcke:


WR
DBIN
INT
INTE
HOLD
HOLD-ACK
WAIT
READY
SYNC
RESET
ϕ_1
ϕ_2
zeitliche und logische Steuerung
Akkumulator 8bit
Akkumulator/ Signalspeicher 8bit
Dezimal-Korrektur
ALU Arithmetische-Logik-Einheit 8bit
Kennzeichnung 5bit
Kennzeichen-Signalspeicher 8bit
Befehlsdecoder und Kontrolle
Befehlsregister
Multiplexer
Registerauswahl
W-REG 8bit
Z-REG 8bit
B-REG 8bit
C-REG 8bit
D-REG 8bit
E-REG 8bit
H-REG 8bit
L-REG 8bit
Stapelzeiger 16bit
Befehlszähler 16bit
Adressen-Speicher 16bit
interner Daten-Bus
interner Adressen-Bus
Datenbus-Puffer
Adressen-Puffer
D_0 D_1 D_2 D_3 D_4 D_5 D_6 D_7
Daten-Bus
A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15}
Adressen-Bus

Register-Matrix und Adressen-Logik

Dieser Block befindet sich auf der rechten Seite von Bild 2.1.1 und besteht aus 13 Teilen:

Befehlszähler (program counter, PC) mit 16 Bits,
Stapelzeiger (stack pointer, SP) mit 16 Bits,
Register-Teil mit acht Registern, B-, C-, D-, E-, H- und L-Register mit je 8 Bits und zwei Zwischenregister W- und Z-Register, mit je 8 Bits. Die Register sind paarweise gruppiert und sind für allgemeine Anwendungen.

In dem Befehls- oder Programm-Zähler (program counter) sind die Speicheradressen für die einzelnen Programmbefehle gespeichert. Ein Programm setzt sich in einem Mikroprozessor immer aus Befehlen zusammen, die im Befehls-Speicher oder Befehls-Register gespeichert sind. Der Befehls-Zähler adressiert nun den Befehls-Speicher, um so festzustellen, welche Operation durchzuführen ist.
Der Speicher-Zähler beginnt bei dem Zählerstand 0 und gibt einen Befehlsabruf an den Befehls-Speicher ab und zählt dann um 1 weiter. Nach einem Freigabe-Signal gibt der Speicher-Zähler einen neuen Befehlsabruf auf den Befehls-Speicher ab und zählt um 1 weiter. Wir haben damit den Zählerstand 2 im Befehls-Zähler. Da der Befehlszähler 16 Bits hat, ergibt sich ein Inhalt von 65 536 Befehlsabrufen für den 8080. Wir können in diesem Befehls-Zähler insgesamt 65 536 beliebige Befehle abrufen lassen.

Der Stapelzeiger (Stack Pointer) enthält die Adresse der nächsten Stapelstelle im Speicher. Der Stapelzeiger wird mittels dem Programm auf einen bestimmten Anfangswert gesetzt. Normalerweise beginnen wir bei 0, aber wir haben die Möglichkeit durch die Bestimmung des Anfangswertes jeden Bereich des Schreib-Lese-Speichers als Stapel zu verwenden. Durch die Adresse des Befehls-Zählers läuft das Hauptprogramm ab, das im RAM gespeichert ist. Bei einer Adresse wird das Hauptprogramm unterbrochen und eine Subroutine, ein Unterprogramm eingebaut. Das Unterprogramm ist ein Sprung-Befehl (jump) zu einem anderen Teil des Speichers. Ein Sprungbefehl enthält immer die Adresse des darauffolgenden Befehls.

Dieser folgende Befehl darf sich in einer beliebigen Speicheradresse befinden, sofern nur der programmierte Sprung die richtige Adresse angibt. Während des Ablaufes des Sprungbefehls ersetzt der Prozessor den Inhalt seines Programm-Zählers durch die Adresse, die in dem Sprungbefehl angegeben ist. Auf diese Art bleibt uns das Hauptprogramm erhalten.

Wir rufen (CALL) ein Unterprogramm ab. Bei diesem Sprung-Befehl hält der Mikroprozessor den Inhalt des Programm-Zählers fest. Dadurch können wir anschließend an das Unterprogramm sofort wieder an der richtigen Stelle unseres Hauptprogrammes beginnen. Das Unterprogramm stellt dadurch ein eigenständiges Programm innerhalb des Hauptprogrammes dar.
Das Unterprogramm besteht aus einer Vielzahl von Befehlen, die wir sonst im Hauptprogramm immer wiederholen müssen. Selbst bei kleineren Hauptprogrammen wird daher das Unterprogramm bereits eingesetzt. Funktionen, die oft im Unterprogramm geschrieben werden, sind die Berechnung eines Sinus oder Cosinus

oder der Logarithmus einer Variablen. Unterprogramme werden vom Mikroprozessor so abgewickelt, daß ihm eine ordnungsgemäße Rückkehr in das Hauptprogramm möglich ist.
Wenn der Mikroprozessor einen Rufbefehl erhält, erhöht er die Zahl im Befehls- oder Programm-Zähler um 1 und speichert den Inhalt des Zählers in einem hierfür reservierten Speicherplatz mit der Bezeichnung Stapel-Speicher (stack) ab. Der Stapel -Speicher speichert damit die nächste Adresse ab, wenn das Unterprogramm beendet ist. Dadurch wird sofort nach Beendigung des Unterprogrammes im Hauptprogramm weiter gearbeitet. Der Mikroprozessor braucht seinen Einstieg in das Hauptprogramm nicht erst umständlich zu suchen. Der Stapel-Speicher hebt so die erste Adresse nach Beendigung des Unterprogrammes auf. Dadurch ergeben sich wesentlich höhere Geschwindigkeiten bei einem Programmablauf. Anschließend lädt der Mikroprozessor die Adresse, die in dem CALL-Befehl enthalten ist, in seinen Befehls- oder Programmzähler.

Der nächste eingeholte Befehl ist damit der erste Schritt in das Unterprogramm. Der letzte Befehl aus dem Unterprogramm ist der Rückkehr-Befehl (RETURN), der aber keine Adresse enthalten braucht. Erhält der Mikroprozessor seinen Rückkehr-Befehl aus dem Unterprogramm in das Hauptprogramm, so ersetzt er einfach den bestehenden Inhalt seines Programm-Zählers durch die im Stapel-Speicher befindliche oberste Adresse. Durch die Setzung auf die oberste, also auf die letzte Adresse, wird der Mikroprozessor zur Wiederaufnahme seines Hauptprogrammes veranlaßt und zwar unmittelbar.
Wir können durch den CALL- und RETURN-Befehl Unterprogramme ineinander verschachteln. Mit einem CALL gehen wir in das erste Unterprogramm. Ist dieses beendet gehen wir mittels RETURN aus dem Unterprogramm, aber auf der nächsten Adresse im Hauptprogramm steht CALL für das nächste Unterprogramm. Eine Programmverschachtelung dieser Art ist möglich, solange der Mikroprozessor genügend Kapazität für die Speicherung der notwendigen RETURNs, sowie die erforderliche Logik besitzt. Die maximale Größe einer Verschachtelung ist von der Größe des Stapelspeichers abhängig. Hat ein Stapelspeicher die Kapazität für drei Rückkehr-Befehle, so kann die Verschachtelung von Unterprogrammen erfolgen. Der 8080 verwendet einen reservierten Teil des RAM-Speichers als Stapelspeicher und verwaltet intern lediglich ein sogenanntes Stapelzeiger-Register (Stack Pointer), das dann die jeweilige Adresse der letzten Stapelspeicher-Einschreibung enthält.
Die Verschachtelung von Unterprogrammen in einem externen Stapelspeicher ist normalerweise unbegrenzt.
Der Stapelzeiger wird jedesmal hochgezählt, wenn Daten in den Stapel eingegeben werden und hochgezählt, wenn Daten aus dem Stapel entnommen werden. Das Dateneingeben bezeichnen wir als PUSH und das Entnehmen als POP. Das bedeutet für den Stapel ein wachsen nach unten.

In Bild 2.1.1 befinden sich im Register noch sechs allgemeine Register. Diese Register können wir entweder als selbständige 8-Bit-Register oder auch als 16-Bit-Registerpaare verwenden. Die beiden Zwischenspeicher-Register W und Z sind nicht durch das Programm adressierbar und sie werden daher nur für die interne Ausfüh-

rung von Befehlen benutzt. 8-Bit-Worte aus dem Befehls-BUS können über den Registerauswahl-Multiplexer zwischen dem internen Daten-BUS und der Register-Matrix übertragen werden. 16-Bit-Übertragungen sind zwischen der Register-Matrix und dem Adressen-Signal-Speicher oder dem Auf/Abwärts-Zähler möglich. Der Adressen-Signal-Speicher empfängt die Daten von einem der drei Registerpaare und steuert damit den 16-Adressen-Buffer (A_0 bis A_{15}), sowie auch den Auf/Abwärts-Zähler. Der Auf/Abwärts-Zähler erhält seine Daten vom Adressen-Signal-Speicher und überträgt sie zu der Register-Matrix. Die 16-Bit-Daten können entweder hinauf- oder herunter gezählt oder einfach zwischen den Registern übertragen werden.

Arithmetik-Logik-Einheit (ALU)

Die ALU bildet den Mittelpunkt des 8080 und enthält folgende Register:

Akkumulator mit 8-Bit,
Zwischenspeicher-Akkumulator mit 8-Bit (ACT)
Zustandsregister mit 5-Bit (flag register) für Null, Übertrag, Parität und Hilfs-Übertrag
Zwischenspeicher-Register mit 8-Bit (TMP)

Alle rechentechnischen und logischen Probleme werden in der ALU durchgeführt, wie arithmetische, logische und umlaufende Operationen. Die ALU erhält die Daten von dem vorgeschalteten Zwischenspeicher-Register (TMP), dem Zwischenspeicher-Akkumulator (ACT) und dem Übertrags-Flip-Flop. Das Ergebnis der Rechenoperation kann entweder zum internen Daten-Bus oder zum Akkumulator übertragen werden. Von der ALU können wir auch die Daten an das Kennzeichenregister mit 5-Bits übertragen.
Die ALU besteht aus einem Addierer, mit dessen Hilfe wir den Inhalt zweier Register addieren können. Durch die entsprechende Programmierung können wir auch Routinen (Unterprogramme) für das Subtrahieren, Multiplizieren und Dividieren durchführen. Die Routine führen wir ein, damit das Hauptprogramm nicht belastet wird.
Neben den arithmetischen Rechenoperationen können wir auch mit der ALU logische Funktionen, wie die Daten komplementieren, negieren oder um eine Stelle verschieben, ausführen.
In der ALU sind sogenannte Zustands-Bits, die Flags enthalten. Die Flags geben uns über die Ergebnisse der arithmetischen und logischen Operationen Aufschluß. Die Flags geben außerdem Informationen ab, wenn ein Übertrag, eine Null, ein Vorzeichen oder eine Parität auftritt.

Übertrag	=	carry
Null	=	zero
Vorzeichen	=	sign
Parität	=	parity
Hilfs-Übertrag	=	auxiliary-carry

Wir können Programm-Sprünge mit den Flags durchführen. Hierzu führt das Programm einen bestimmten Sprung aus, wenn die Flags eine Kombination aufweisen. Wir können das Programm in eine Routine (Unterprogramm) übergehen lassen, wenn z.B. ein Übertrag bei einer Addition auftritt. In der Routine würde z.B. stehen, addiere bei einem Übertrag die Summe mit einer dualen 6. Dadurch erhalten wir eine Summe im BCD-Code. Das Zwischenspeicher-Register (TMP) nimmt die Informationen des internen Daten-Busses auf und übergibt sie ganz oder nur teilweise an die ALU oder an das Zustands-Register. Der Akkumulator (ACC) kann von der ALU und vom internen Daten-Bus seine Informationen erhalten. Der Akkumulator überträgt seine Ergebnisse zum Zwischenspeicher-Akkumulator (ACT) und an den internen Daten-Bus. Der Inhalt des Akkumulators (ACC) und des Hilfs-Übertrags-Flip-Flops kann während der Durchführung eines DAA-Befehls (Akkumulator -Inhalt wird in eine zweistellige Zahl umgewandelt) auf die Dezimalkorrektur geprüft werden.

Befehls-Register und Steuer-Logik

Das Befehls-Register bezieht aus dem Daten- und Adressen-BUS sein Informationen. Normalerweise beginnt ein Programm mit der Adresse 0. Über den internen Daten-BUS kann das Befehls-Register seinen ersten Befehl aufnehmen, speichern und diesen an den nachfolgenden Befehlsdecoder und Operationszyklusdecoder abgeben.
Der Ausgang des Befehlsdecoders liefert uns mit verschiedenen Zeitimpulsen die Steuersignale für die Register-Matrix, die ALU und für die Daten-Buffer. Die Steuersignale steuern darüberhinaus noch mit den externen Steuersignalen zusammen den gesamten Steuerbereich der Schaltung an und liefern die Zustands- und Zyklus-Zeitsignale.
Über den internen Daten-Bus liegt ein 8-stelliges Wort an dem Befehlsregister an. Aus diesem 8-stelligen Wort kann der Mikroprozessor seine 78 Befehle entnehmen. Normalerweise können wir mit einem 8-stelligen Wort 256 Befehle übertragen, bei der 8080 werden aber nur 78 Befehle ausgeführt. Der Mikroprozessor liest in zwei Operationen einen Befehl (instruction-FETCH) ein. Zuerst sendet er die in seinem Befehlszähler befindliche Adresse zum Speicher. Dieser übermittelt daraufhin das adressierte Byte zum Mikroprozessor, der das empfangene Befehlswort in dem Befehls-Register speichert und es zur Steuerung der Abläufe während der nachfolgenden Befehlsausführung verwendet.

Die Umwandlung eines Befehlswortes in einen Programmablauf wird durch den Befehls-Decoder und den damit verbundenen Steuerschaltungen durchgeführt. Das Befehlswort wird von dem Befehls-Register gespeichert und von dem Befehls-Decoder decodiert. Damit kann dann der Mikroprozessor arbeiten.
In den meisten Anwendungen reicht ein 8-Bit-Befehlswort für eine Rechenoperation aus. In der Praxis kann es aber vorkommen, daß die Befehlsausführung mehrere Informationen benötigt, als wir in 8-Bits ausdrücken können. Dieser Fall tritt auf, wenn ein Befehlswort eine Operation und einen Speicherplatz beinhalten muß. Das einzelne Befehlswort kann die Operation durchführen, kann aber nicht

die Adresse aufnehmen. Für diesen Fall müssen wir den Befehl aus 2 oder 3 Bytes zusammenstellen.

Ein Byte-Befehl: Bei einem Ein-Byte-Befehl handelt es sich meistens um einen Operationsbefehl, wie z.B. arithmetische-, logische-, Speicherungs-, Routineanfang oder Routineende-Befehle.

z.B.:

CMC-Befehl = Carry-Bit soll negiert werden

Zwei Byte-Befehl:- Bei einem Zwei-Byte-Befehl handelt es sich meistens um einen Operationsbefehl mit der Adresse für den Operanden.
z.B.:

JPO-Befehl = Bei Paritäts-Bit springe auf

ADR = Adresse 15

Wir haben hier einen Befehl für einen unmittelbaren Operationshinweis oder eine I/O-Operation (I = input, 0 = output, Ein/Ausgabe-Befehl)

Drei Byte-Befehl: Die Adresse bei dem Zwei-Byte-Befehl reicht nur bis zur 255. Adresse. Mit der 8080 können 65 536 Adressen angesteuert werden. Hierzu ist daher ein Drei-Byte-Befehl notwendig.
z.B.:

JPO-Befehl = Bei Paritäts-Bit springe auf

nADR = niederwertige Adresse 140

hADR = höherwertige Adresse 16 384

Die gesamte Adresse lautet: 16 384 + 140 = 16 524
Bei einem Paritäts-Bit springt der Mikroprozessor auf diese Adresse.

Aufeinanderfolgende Befehlsworte befinden sich im Speicher in hintereinanderliegenden Speicherplätzen und der Mikroprozessor erhält den vollen Befehl durch den 2 oder 3 FETCH-Zyklus. Unter einem FETCH verstehen wir einen Abruf. Das erste vom Speicher eingeholte Wort wird in dem Befehls-Register eingespeichert, das zweite Wort in einem Zwischenspeicher-Register, ebenso anschließend das dritte Wort. Nach der Abspeicherung der Worte beginnt der Mikroprozessor zu arbeiten.

Die Steuer-Logik des Mikroprozessors ist für den Ablauf eines Programmes erforderlich. Zusammen mit den Taktsignalen sorgt sie für den ordnungsgemäßen Ablauf des Programmes. Wird ein Befehl eingeholt und decodiert, so veranlaßt die

Steuer-Logik durch das Aussenden eines Befehles an alle internen und externen Einheiten des Mikroprozessor-Systems die Aufnahme oder den Beginn einer Operation.
Die Steuer-Logik kann auch Befehle aufnehmen, z.B. eine Unterbrechung des Programmes oder eine Warte-Aufforderung. Bei einer Unterbrechungs-Aufforderung veranlaßt die Steuer-Logik eine Unterbrechung des Hauptprogrammes für die Bevorzugung eines Unterprogrammes. Nach der Beendigung eines Unterprogrammes gibt die Steuer-Logik das Hauptprogramm wieder frei. Eine Warte-Aufforderung (WAIT) wird oft von einem externen Speicher oder einer I/O-Einrichtung erteilt. Die Warte-Aufforderung kommt daher zustande, daß die externen Einrichtungen häufig langsamer als der Mikroprozessor arbeiten. Durch die Steuer-Logik bleibt der Mikroprozessor solange stehen, bis die externe Einrichtung die Daten aufgenommen oder abgegeben hat.

Verbindungen im Mikroprozessor

Die Verbindungen im Mikroprozessor sind der Daten-BUS, der Adressen-BUS und der Kontroll-BUS. Über den Daten-BUS erhalten die einzelnen Einrichtungen im System ihre Daten. Da der 8080 einen Daten-BUS von 8-Bits hat, können wir ein achtstelliges Datenwort mit 256 Möglichkeiten für Befehle und Informationen transportieren.
Über den Adressen-BUS steuern wir die einzelnen Einrichtungen im System an. Bei der 8080 hat der Adressen-BUS eine 16-Bit-parallele Struktur. Durch diese Struktur können insgesamt 65 536 Adressen angesteuert werden. Bei der Programmierung muß aber darauf geachtet werden, damit die Adressen nicht weit voneinander liegen. Jeder Programmsprung auf eine weit entfernte Adresse kostet Zeit!
Über den Kontroll-BUS erfolgt die Überwachungsfunktion der Steuer-Logik. Die Steuer-Logik ist über einen 6-Bit parallelen Bus mit den internen Einrichtungen verbunden.

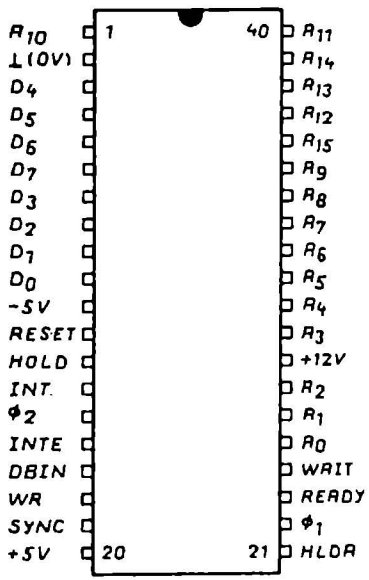


Bild 2.1.2 Anschlußbelegung des Mikroprozessors 8080 (Draufsicht)

Die Daten für den Mikroprozessor kommen über den Daten-Buffer. Diese Buffer-Einrichtung ist für den Zweiweg-Betrieb geeignet, d.h. über den Buffer kann eingeschrieben, aber auch ausgelesen werden. Der Daten-Buffer ist die Schnitt-Stelle zwischen dem internen Daten-BUS und dem externen Daten-BUS. Bei einer Ausgabe-Operation wird der Inhalt des internen Daten-Busses zunächst in einen 8-Bit-Signal-Speicher geladen, der anschließend die Ausgangsbuffer des Mikroprozessors ansteuert. Die Ausgangsbuffer sind während einer Daten-Aufnahme und außerhalb einer Übetragungs-Operation ausgeschaltet. Bei der Eingabe von Daten von dem externen zum internen Daten-BUS wird zu Beginn jedes internen Operationsschrittes der interne BUS geladen, aber nicht, wenn ein Übertragungsschritt erfolgt.

Befehls-Liste für den 8080:

Der Mikroprozessor 8080 kann mittels der Befehls-Liste arbeiten. Bei der Programmierung ergeben sich Symbole und Abkürzungen:
DDD oder SSS: Durch ein Bit-Muster können wir die einzelnen Register ansteuern. DDD = Zielstelle und SSS = Quelle. Es gilt folgende Tabelle:

L L L	Register B
L L H	Register C
L H L	Register D
L H H	Register E
H L L	Register H
H L H	Register L
H H H	Register A (Akkumulator)

r_1, r_2: Bezeichnung für eine der Register A, B, C, D, E, H und L.

Transfer-Befehle:

Register→Register

MOV r_1 r_2

L	H	D	D	D	S	S	S

L L L	L L L	Register	B
L L H	L L H	"	C
L H L	L H L	"	D
L H H	L H H	"	E
H L L	H L L	"	H
H L H	H L H	"	L
H H L	H H L	Speicher	M
H H H	H H H	Register	A

Lade Register r_1 mit Inhalt des Registers r_2. Hier erfolgt eine Datenübertragung zwischen zwei Registern. Mit dem S-Bit-Muster bestimmen wir die Registerstelle und mit dem D-Bit-Muster die Zielstelle.

XCHG

H	H	H	L	H	L	H	H

Die beiden Register H und L (Registerpaar H und L) werden mit dem Registerpaar D und E vertauscht. Hierbei werden zwei 8-Bit-Register zu einem 16-Bit-Register zusammengeschaltet. Wir tauschen 16 Bits aus.

XTHL

H	H	H	L	L	L	H	H

Der Inhalt des L-Registers wird mit dem Inhalt des Speicherplatzes, der durch den Kellerspeicher adressiert wird ausgetauscht. Der Inhalt des H-Registers wird mit dem Inhalt des Speicherplatzes SP+1 ausgetauscht.

SPHL

H	H	H	H	H	L	L	H

Die 16-Daten-Bits der beiden Registerpaare H und L ersetzen den Inhalt des Kellerspeichers. Dabei bleibt aber der Registerinhalt des Registerpaares H und L unverändert.

Speicher, Peripherie Register :

MOV r_1 M

L	H	D	D	D	H	H	L

L	L	L
L	L	H
L	H	L
L	H	H
H	L	L
H	L	H
H	H	H

Register	B
"	C
"	D
"	E
"	H
"	L
"	A

Mit dem Befehl wird der Inhalt einer Speicherstelle in das Register r_1 vorgenommen. Dieser Datentransfer aus einer Speicheradresse wird durch den Inhalt der beiden Register H und L spezifiziert. Das Register L speichert die acht wertniedrigen und das Register H die acht werthöheren Adressen.

LDA adr

L	L	H	H	H	L	H	L

Der Inhalt eines Speicherplatzes wird in den Akkumulator (Register A) geladen. Bei einer Adressierung eines Speicherplatzes wird der gesamte Inhalt des Speicherplatzes ersetzt.

LDAX rp1 | L | L | R | P | H | L | H | L |

L L	für Register B und L
L H	" " D u. E
H L	" " H u. L
H H	für die Flags und Register A

Der Inhalt des Speicherplatzes, der durch das Registerpaar (rp oder RP) B und C bzw. D und E adressiert ist, wird in den Akkumulator übertragen.

LHLD adr | L | L | H | L | H | L | H | L |

Hier wird das Registerpaar H und L direkt mit dem Inhalt der Adressen adr geladen. Das Adressen-Byte ersetzt den Inhalt des Registers L. Das Byte mit der nächsthöheren Adresse adr+1 ersetzt den Inhalt des H-Registers.

Beim POP wird der Inhalt des Kellerspeichers in ein Register übernommen. In das spezifizierte Registerpaar wird der Inhalt zweier Speicherbyte geladen. Es gilt:

POP rp1 | H | H | R | P | L | L | L | H |

L L	für Register B und C
L H	für Register D und E
H L	für Register H und L
H H	für die Flags und Register A

Das Daten-Byte mit der Adresse für den Rücksprung ist identisch mit dem Kellerspeicher und es wird in das spezifizierte Registerpaar geladen. Das Daten-Byte mit der Adresse SP+1 (Stack Pointer + 1) wird in das erste Register des Registerpaares geladen. Wird ein Registerpaar spezifiziert, ersetzt das Byte, das durch den Kellerspeicher + 1 adressiert wird, die Bedingungs-Bits. Nachdem die Daten geladen sind, wird der Kellerspeicher um 2 erhöht.

IN nr | H | H | L | H | H | L | H | H |

Hier wird ein Daten-Byte von einem Eingabe-Baustein eingeschrieben. Die Nummer des Bausteines kann zwischen 0 und 255 liegen. Für den Betrieb ist ein expliziter Operand erforderlich. Dieser wird einfach an das Operations-Code-Byte (IN-Befehl) angehängt. In dem Operanden-Byte befindet sich die Eingabe-Baustein-Nummer. Das Operanden-Byte ist im Dual-Code aufgebaut, daher die explizite Form für das Byte.

Konstante→ Registerpaar :

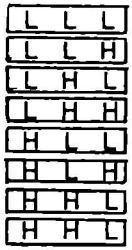
LXI rp1 adr | L | L | R | P | L | L | L | H |

L L	für Register B und L
L H	" " D " E
H L	" " H " L
H H	für die Flags und Register H

Hier wird eines der Registerpaare mit einer Konstanten geladen. Das dritte Byte dieses Befehls wird in das erste Register des spezifizierten Registerpaares geladen, während das zweite Byte des Befehls in das zweite Register geladen wird. Ist das Registerpaar SP (Stack Pointer) spezifiziert, ersetzt das zweite Byte des Befehls die acht niederwertigen Bits des Kellerspeichers und das dritte Byte ersetzt die acht höherwertigen Bits des Kellerspeichers.

Register Speicher, Peripherie :

MOV M r1



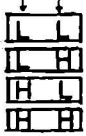
Register B
" C
" D
" E
" H
" L
Speicher M
Register A

Dieser Befehl bringt die Daten aus dem Register r1 an einen Speicherplatz im Schreib-Lese-Speicher (RAM), dessen Adressen durch den Inhalt der beiden Register H und L spezifiziert sind. Das Register L beinhaltet dabei die niederwertige Adresse und das Register H die höherwertige Adresse.

STA adr

Der Akkumulator-Inhalt wird direkt abgespeichert. Der Inhalt ergibt ein Byte. Mit den Adressen-Bytes wird der Speicherplatz direkt festgelegt.

STAX rp1



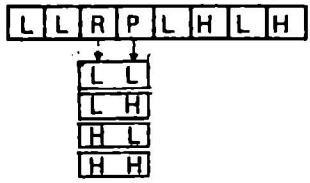
für Register B und L
" " D " E
" " H " L
für die Flags und Register A

Der Inhalt eines Speicherplatzes, der durch das Registerpaar B und C oder D und E adressiert wird, wird durch den Inhalt des Akkumulators überschrieben. Damit wird der Inhalt des Akkumulators ausgelesen und auf einen Speicherplatz eingeschrieben.

SHLD adr

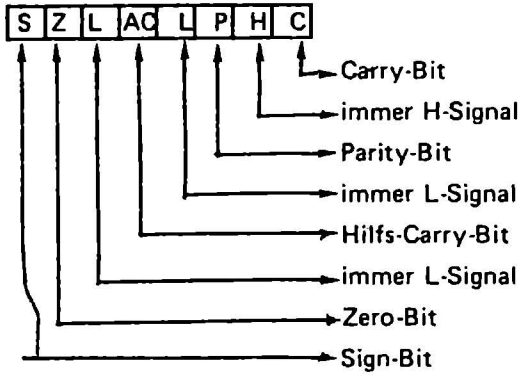
Hier wird der Inhalt der beiden Register H und L direkt abgespeichert. Der Inhalt des L-Registers ist ein Byte, das von zwei Adressen-Bytes adressiert wird, auf einem Speicherplatz abgelegt und das H-Register wird im nächsthöheren Byte gespeichert.

PUSH rp1



für Register B und L
" " D " E
" " H " L
für die Flags und Register A

Mit dem PUSH-Befehl wird die letzte Adresse vor dem Sprung in ein Unterprogramm festgehalten. Der Inhalt des spezifizierten Registerpaares wird in jenen zwei Speicher-Bytes sichergestellt, die durch den Kellerspeicher SP adressiert werden. Der Inhalt des ersten Registers, d.h. Register B,D,H oder die Zustands-Flip-Flops werden in dem Byte mit der Adresse des Kellerspeichers -1 abgespeichert, der Inhalt des zweiten Registers, d.h. Register C, E, L oder A, wird in dem Byte mit der Adresse des Kellerspeichers -2 abgespeichert. Es ergibt sich folgendes Format:



Sind die Daten sichergestellt, so wird der Kellerspeicher immer um -2 verringert.

OUT

Hier wird ein Daten-Byte auf einen Ausgabe-Baustein geschaltet. Der Inhalt des Akkumulators kann über den Ausgabe-Baustein angezeigt werden. Für die Daten-Ausgabe ist ein explizites Operand-Byte erforderlich. Mit dem Operand, der im Dual-Code programmiert ist, können von maximal 256 Ausgabe-Einheiten eine angesteuert werden.

Konstante →Register-Speicher :

MVI M konst

Dieser Befehl gestattet eine unmittelbare Ladung auf einen Speicherplatz, der durch den Inhalt des Registerpaares H und L adressiert worden ist. Das Register H beinhaltet die wertniedrige und das Register L die werthöhere Adresse. Die Konstante kann einen Wert zwischen 0 und 255 aufweisen.

MVI r1 konst

L	L	D	D	D	H	H	L

L L L	Register B
L L H	" C
L H L	" D
L H H	" E
H L L	" H
H L H	" L
H H H	" A

Eines der sieben Register kann mit einer Konstanten geladen werden. Mit dem D kennzeichnen wir die Zielstelle, also das Register. Die Konstante kann einen Wert zwischen 0 und 255 aufweisen.

Befehle für arithmetische Operationen :

Für folgende Register wird der Inhalt um eine 1 erhöht :

INR r1

L	L	D	D	D	H	L	L

L L L	Register B
L L H	Register C
L H L	Register D
L H H	Register E
H L L	Register H
H L H	Register L
H H L	Speicherzugriff M
H H H	Register A

Der Inhalt eines Registers wird incrementiert.

INR M

L	L	H	H	L	H	L	L

Der Inhalt des durch das Registerpaar H und L adressierten Bytes wird um eine 1 erhöht (incrementiert).

Für folgende Register wird der Inhalt um eine 1 verringert:

DCR r1

L	L	D	D	D	H	L	H

L L L	Register B
L L H	Register C
L H L	Register D
L H H	Register E
H L L	Register H
H L H	Register L
H H L	Speicherzugriff M
H H H	Register A

Der Inhalt eines der Register wird incrementiert;

DCR M

Der Inhalt des durch das Registerpaar H und L adressierten Bytes wird um eine 1 verringert (decrementiert).

Der Inhalt der folgenden spezifizierten Registerpaare wird um eine 1 erhöht:

INX rp1

L L	Registerpaar B und E
L H	Registerpaar D und E
H L	Registerpaar H und L
H H	Registerpaar SP

Der Inhalt eines Registerpaares hat 16 Bits

Der Inhalt der folgenden spezifizierten Registerpaare wird um eine 1 verringert:

DCX rp1

L L	Registerpaar B und E
L H	Registerpaar D und E
H L	Registerpaar H und L
H H	Registerpaar SP

Der Inhalt eines Registerpaares hat 16-Bits.

ADD r1

Ein spezifiziertes Byte eines der sieben Register wird zum Inhalt des Akkumulators addiert.

ADD M

Soll ein Byte des Speichers über die gespeicherte Adresse von dem Registerpaar H und L spezifiziert werden, so erfolgt eine Addition zum Inhalt des Akkumulators.

ADC r1

Ein spezifiziertes Byte eines der sieben Register wird mit Carry-Bit zum Inhalt des Akkumulators addiert.

ADC M

Soll ein Byte des Speichers über die gespeicherte Adresse H und L spezifiziert werden, so erfolgt eine Addition mit Carry-Bit zum Inhalt des Akkumulators.

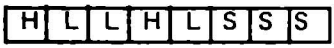
Eine 16-Bit-Zahl im folgenden Registerpaar wird zur 16-Bit-Zahl im Registerpaar H und L addiert.

DAD rp1

L L	Registerpaar B und C
L H	Registerpaar D und E
H L	Registerpaar H und L
H H	Registerpaar SP

Das Ergebnis der Addition steht im Registerpaar H und L. Bei einer Addition von Registerpaar H und L mit dem gleichen Registerpaar H und L, wird der Inhalt verdoppelt.

SUB r1



Ein spezifiziertes Byte eines der sieben Register wird unter Verwendung des B-Komplements vom Inhalt des Akkumulators subtrahiert. Entsteht bei der Subtraktion der höchsten Bitstelle kein Überlauf, so erfolgt die Setzung des Carry-Bits.

SUB M

Soll ein Byte des Speichers über die gespeicherte Adresse von dem Registerpaar H und L spezifiziert werden, so erfolgt eine Subtraktion vom Inhalt des Akkumulators.

SBB r1

Das Carry-Bit wird intern zu dem Inhalt des spezifizierten Bytes dazuaddiert. Der Inhalt wird dabei negiert und vom Akkumulator subtrahiert. Es entsteht eine Subtraktion nach dem B-1-Komplement.

SUB M

Soll ein Byte des Speichers über die gespeicherte Adresse von dem Registerpaar H und L spezifiziert werden, so erfolgt die Subtraktion vom Inhalt des Akkumulators nach dem B-1-Komplement, also mit Carry-Bit.

ADI konst

Ein Daten-Byte wird zu dem Inhalt des Akkumulators hinzuaddiert. Das Daten-Byte ist eine Konstante zwischen 0 und 255.

ACI konst

Ein Daten-Byte mit einem Carry-Bit wird zu dem Inhalt des Akkumulators hinzuaddiert. Das Daten-Byte ist eine Konstante zwischen 0 und 255.

SUI konst

Ein Daten-Byte wird von dem Inhalt des Akkumulators subtrahiert. Das Daten-Byte ist eine Konstante zwischen 0 und 255.

SBI konst

Ein Daten-Byte mit einem Carry-Bit wird von dem Inhalt des Akkumulators subtrahiert. Das Daten-Byte ist eine Konstante zwischen 0 und 255.

DAA

Mit dem Befehl kann der Akkumulator korrigiert werden. Wir haben dann eine Dezimalkorrektur des Akkumulators. In dem Akkumulator befindet sich ein Byte, das in zwei 4-Bit-lange-Wörter umgewandelt werden kann. Ist das wertniedrigere 4-Bit-Wort größer als 9 oder das Hilfs-Carry-Bit ist gesetzt, wird zu dem Inhalt des Akkumulators eine duale 6 (LHHL) hinzuaddiert. Andernfalls bleibt der Inhalt des Akkumulators unverändert.
Wenn nach der Korrektur sich das höherwertige 4-Bit-Wort verändert hat, größer als 9 oder das Carry-Bit gesetzt ist, wird zu dem höherwertigen 4-Bit-Wort eine duale 6 (LHHL) in dem Akkumulator hinzuaddiert.
Dieser Befehl ist nur erforderlich, wenn Dezimalzahlen addiert werden sollen. Es ist der einzige Befehl, der durch das Hilfs-Carry-Bit beeinflußt wird.

Befehle für logische Operationen:

CMA

Durch den Befehl wird der gesamte Inhalt des Akkumulators komplementiert. Aus L-Signalen wird H und aus H wird L.

ANA r1

Das spezifizierte Byte aus den internen Registern wird mit dem Inhalt des Akkumulators konjunktiv verknüpft. Das UND-Gatter wirkt dabei immer bitweise.

ANA M

Konjunktive Verknüpfung eines spezifizierten Bytes, die durch den Inhalt des Registerpaares H und L adressiert worden ist, mit dem Akkumulator

ANI konst

Der Inhalt des Akkumulators wird mit einer Konstanten konjunktiv verknüpft.

ORA r1

Das spezifizierte Byte aus den internen Registern wird mit dem Inhalt des Akkumulators verknüpft. Das ODER-Gatter wirkt dabei immer bitweise.

ORA M

Disjunktive Verknüpfung eines spezifizierten Bytes, das durch den Inhalt des Registerpaares H und L adressiert worden ist, mit dem Akkumulator

ORI konst

Der Inhalt des Akkumulators wird mit einer Konstanten disjunktiv verknüpft. Die Konstante kann einen Wert zwischen 0 und 255 aufweisen.

XRA r1

Das spezifizierte Byte aus den internen Registern wird mit dem Inhalt des Akkumulators durch eine Exclusiv-ODER-Verknüpfung verbunden. Die Verknüpfung wirkt dabei immer bitweise.

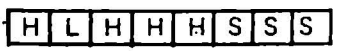
XRA M

Exclusiv-ODER-Verknüpfung eines spezifizierten Bytes, das durch den Inhalt des Registerpaares H und L adressiert worden ist, mit dem Akkumulator.

XRI konst

Der Inhalt des Akkumulators wird mit einer Konstanten über ein Exclusiv-ODER verknüpft. Die Konstante kann einen Wert zwischen 0 und 255 aufweisen.

CMP r1



Das spezifizierte Byte der internen Register wird mit dem Inhalt des Akkumulators verglichen. Der Vergleich wird durch eine interne Subtraktion von dem Register-Inhalt und dem Akkumulator durchgeführt. Dabei wird der Register-Inhalt von dem Inhalt des Akkumulators subtrahiert. Beide Inhalte bleiben aber unverändert. Es wird in Abhängigkeit des Resultates ein Bedingungs-Bit gesetzt. Das Zero-Bit setzt sich, wenn der Inhalt des Akkumulators mit dem Inhalt des Registers identisch ist. Das Zero-Bit wird zurückgesetzt, wenn dies nicht der Fall ist. Die beiden Inhalte sind damit ungleich. Durch die Subtraktion wird das Carry-Bit gesetzt, wenn kein Überlauf aus der höchsten Bitposition entsteht, d.h. wenn der Inhalt des Registers größer ist, als der des Akkumulators. Ist der Inhalt des Akkumulators größer, wird das Carry-Bit zurückgesetzt. Es ergibt sich ein Größer-Kleiner-Vergleicher.

In Verbindung des Zero-Bits und des Carry-Bits ergibt sich ein Größer-Gleich-Kleiner-Vergleich bei diesem Befehl.

CMP M

H	L	H	H	H	H	H	L

Der Vergleich wird zwischen einem adressierten Byte und dem Akkumulator intern durchgeführt. Das Registerpaar H und L enthält hierbei die Adresse für dieses Byte. Der Inhalt des adressierten Bytes wird durch eine Subtraktion des Inhaltes vom Akkumulator durchgeführt. Hat das Zero-Bit ein Signal, so sind beide Inhalte gleich. Andernfalls hat das Zero-Bit kein Signal. Ist der Inhalt des Akkumulators kleiner als der Inhalt des Registers, wird durch die Subtraktion das Carry-Bit gesetzt. Ist der Inhalt des Registers kleiner, so setzt sich durch die Subtraktion das Carry-Bit zurück.

CPI konst

H	H	H	H	H	H	H	L

Hier wird der Vergleich zwischen einer Konstanten und dem Akkumulator durchgeführt. Der Inhalt der Konstanten wird von dem Inhalt des Akkumulators subtrahiert. Ist die Konstante kleiner als der Akkumulator, so wird das Carry-Bit bei der Subtraktion zurückgesetzt. Ist die Konstante größer, setzt sich das Carry-Bit. Sind die beiden Inhalte gleich, wird das Zero-Bit gesetzt.

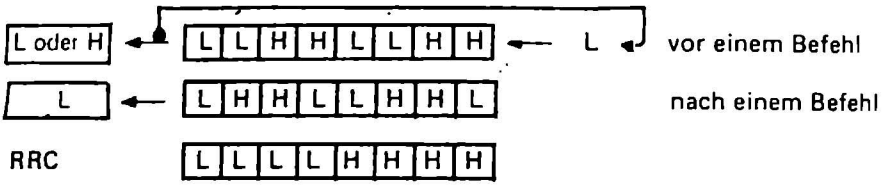
Registeranweisungen.

Akkumulator rotieren:

RLC

L	L	L	L	L	H	H	H

Der Inhalt des Akkumulators wird bei diesem Befehl nach links geschoben. Das Carry-Bit wird gleich nach dem höchstwertigen Akkumulator-Bit gesetzt. Der Inhalt wird pro Befehl um eine Bitstelle nach links geschoben. Das niederwertige Bit des Akkumulators wird mit dem hinausgeschobenen Bit gesetzt. Es ergibt sich folgende Rotationsfunktion.:



Der Inhalt des Akkumulators wird bei diesem Befehl nach rechts geschoben. Das Carry-Bit wird gleich dem niederwertigen Akkumulator-Bit gesetzt. Der Inhalt wird pro Befehl um eine Bitstelle nach rechts geschoben. Das höherwertige Bit des Akkumulators wird mit dem hinausgeschobenen Bit gleichgesetzt. Es ergibt sich folgende Rotationsfunktion:

Carry-Bit	Akkumulator		
L oder H	L L H H L L H H	H	vor einem Befehl
H	H L L H H L L H		nach einem Befehl

RAL

L L L H L H H H		

Der Inhalt des Akkumulators wird um eine Bit-Stelle nach links verschoben. Das Bit, das aus dem Akkumulator geschoben wird, ersetzt das Carry-Bit. Das Carry-Bit ersetzt das niederwertige Bit des Akkumulators. Es ergibt sich folgende Rotationsfunktion:

Carry-Bit	Akkumulator	
L	H L H H L L H H	vor einem Befehl
H	L H H L L H H L	nach einem Befehl

RAR

L L L H H H H H		

Der Inhalt des Akkumulators wird um eine Bit-Stelle nach rechts verschoben. Das Bit, das aus dem Akkumulator geschoben wird, ersetzt das Carry-Bit. Das Carry-Bit ersetzt das höherwertige Bit des Akkumulators. Es ergibt sich folgende Rotationsfunktion.:

Carry-Bit	Akkumulator	
L	H L H H L L H H	vor einem Befehl
H	L H L H H L L H	nach einem Befehl

Übertrags-Bit-Anweisungen:

CMC

L L H H H H H H		

Mit diesem Befehl wird das Carry-Bit negiert, d.h. hat das Carry-Bit ein L-Signal, so wird dies zu einem H und umgekehrt.

STC

L L H H L H H H		

Mit diesem Befehl wird das Carry-Bit auf ein H-Signal gesetzt.

Sprungbefehle:

Unbedingte Sprünge:

PCHL

Mit dem Befehl wird der Befehlszähler geladen. Der Befehlszähler erhält seine Information aus dem Registerpaar H und L. Das Register H hat die acht höherwertigen Bits und das Register L die acht niederwertigen Bits gespeichert. Dies bewirkt daß der Mikroprozessor bei der Befehlsausführung bei dem Befehl sein Programm fortsetzt, der durch das Registerpaar H und L adressiert wird.

JMP adr

Bedingt durch diesen Befehl springt der Mikroprozessor auf die Adresse. Er setzt dann auf dieser Adresse sein Programm fort. Da die Adresse in diesem Fall zwei Bytes benötigt, handelt es sich hierbei um einen Drei-Byte-Befehl.

Bedingte Sprünge:

JC adr

Bei diesem Sprung-Befehl verläßt der Mikroprozessor nur dann sein Programm, wenn das Carry-Bit ein H-Signal hat. Das Programm wird auf der Adresse weitergeführt, die an dem JC-Befehl angehängt ist. Hierbei handelt es sich wieder um einen Drei-Byte-Befehl.

JNC adr

Bei diesem Sprung-Befehl verläßt der Mikroprozessor sein Programm, wenn das Carry-Bit ein L-Signal hat. Das Programm wird auf der angehängten Seite weitergeführt.

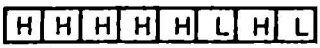
JZ adr

Der Mikroprozessor verläßt sein Programm, wenn das Zero-Bit ein H-Signal aufweist. Das Programm wird auf der angehängten Adresse weitergeführt.

JNZ adr

Hat das Zero-Bit ein L-Signal, verläßt der Mikroprozessor sein Programm und führt es an der angehängten Adresse wieder weiter.

JM adr



Der Mikroprozessor verläßt sein Programm, wenn das Vorzeichen-Bit ein H-Signal hat und führt es an der angehängten Adresse des Befehls weiter.

JP adr

H	H	H	H	L	L	H	L

Bei diesem Sprungbefehl verläßt der Mikroprozessor sein Programm, wenn das Vorzeichen-Bit ein L-Signal hat und führt es an der angehängten Adresse wieder weiter.

JPE adr

H	H	H	L	H	L	H	L

Ist das Resultat einer Rechenoperation gerade, so hat das Paritäts-Bit ein H-Signal. Damit verläßt der Mikroprozessor sein Programm und begibt sich auf die angehängte Adresse bei dem JPE-Befehl.

JPO adr

H	H	H	L	L	L	H	L

Der Mikroprozessor verläßt sein Programm, wenn das Resultat seiner Rechenoperation ungerade ist, d.h. das Paritäts-Bit hat ein L-Signal. Das Programm wird auf der angehängten Adresse vorgesetzt.

RST konst

H	H	N	N	N	H	H	H

Der RST-Befehl ist ein spezieller Befehl für Sprung aus dem Unterprogramm. Der Inhalt des Befehlszählers ist in dem Kellerspeicher abgespeichert und stellt dadurch die Adresse für den Rücksprung aus dem Unterprogramm dar. Der Mikroprozessor arbeitet auf dieser Adresse in seinem Hauptprogramm dann weiter.
Mit den N-Worten kann ein konstanter Rücksprung zwischen 0 und 7 programmiert werden. Hierbei ergeben sich für den Mikroprozessor bis zu acht Unterprogramme, die dann als Interrupt-Behandlungen zu werten sind. Für jede Unterbrechung kann ein RST-Befehl verwendet werden, der aber wiederum einen Unterprogrammsprung für einen speziellen Anwendungsfall sein kann. Das Programm wird durch den RST-Befehl an der Stelle fortgeführt, an der der Unterbrechungs-Befehl aufgetreten ist.

Unterprogramme:

Programmabruf:

CALL adr

H	H	L	L	H	H	L	H

Der CALL-Befehl stellt einen unbedingten Unterprogrammaufruf dar. Das Programm für den Mikroprozessor wird auf der angehängten Adresse weitergeführt. Der Sprung in das Unterprogramm wird auf jeden Fall durchgeführt. Die Rückkehr-Adresse wird im Kellerspeicher gespeichert. Damit kennt der Mikroprozessor seine Rücksprung-Adresse, wenn er das Unterprogramm verläßt.

CC adr

H	H	L	H	H	H	L	L

Der Mikroprozessor springt sofort in sein Unterprogramm, wenn ein Übertrag bei einer Rechenoperation aufgetreten ist. Das Carry-Bit hat hierzu ein H-Signal. Die Adresse vor dem Sprung wird im Kellerspeicher festgehalten.

CZ adr

Ist das Ergebnis im Akkumulator gleich Null, so hat das Zero-Bit ein H-Signal. Der Mikroprozessor verläßt sein Programm und springt in das Unterprogramm. Sein Unterprogramm wird auf der angehängten Adresse weitergeführt. Die letzte Adresse vor dem Sprung ins Unterprogramm wird im Kellerspeicher festgehalten.

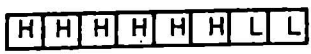
CNC adr

Der Mikroprozessor verläßt sein Programm, wenn das Carry-Bit ein L-Signal hat. Das Programm geht in ein Unterprogramm über und die letzte Adresse wird im Kellerspeicher festgehalten.

CNZ adr

Das Zero-Bit hat ein H-Signal, wenn das Ergebnis des Akkumulators nicht gleich Null ist. Der Mikroprozessor springt ins Unterprogramm und hält die letzte Adresse seines Hauptprogrammes fest. Das Unterprogramm wird auf der angehängten Adresse des CNZ-Befehl weitergeführt.

CM adr



Der Mikroprozessor verläßt bei diesem Befehl sein Hauptprogramm und springt ins Unterprogramm, wenn das Ergebnis des Akkumulators negativ ist, d.h. daß das Sign-Bit ein H-Signal hat. Das Unterprogramm wird auf der angehängten Adresse weitergeführt und die letzte Adresse in dem Kellerspeicher gestapelt.

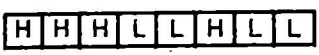
CP adr

Der Unterprogramm-Aufruf gilt für ein positives Ergebnis im Hauptprogramm. Das Sign-Bit hat ein L-Signal und der Mikroprozessor verläßt darauf hin sein Hauptprogramm und führt sein Unterprogramm auf der angehängten Adresse des CP-Befehls weiter.

CPE adr

Hat der Akkumulator eine gerade Parität, dann hat das Parity-Bit ein H-Signal. Es kommt zu dem Aufruf; "springe in das Unterprogramm"; Das Unterprogramm wird auf der angehängten Adresse des CPE-Befehls weitergeführt. Die letzte Adresse des Hauptprogrammes wird in dem Kellerspeicher gestapelt.

CPO adr



Das Unterprogramm wird auf der angehängten Adresse des Befehls weitergeführt, wenn das Parity-Bit ein L-Signal hat, d.h. das Ergebnis oder der Inhalt des Akkumulators hat eine ungerade Parität. Die letzte Adresse wird im Kellerspeicher festgehalten.

Rücksprungbefehle.

RET

Der Befehl ist ein unbedingter Rücksprung aus einem Unterprogramm. Es wird auf jeden Fall ein Rücksprung durchgeführt. Die Befehlsausführung wird mit dem auf dem letzten CALL-Befehl folgenden Befehl fortgesetzt. Mit dem RET wird die letzte Adresse, die im Kellerspeicher sichergestellt worden ist, herausgeholt und in den Befehlszähler gespeichert. Mit dieser rückgespeicherten Adresse kann der Mikroprozessor sein Hauptprogramm wieder aufnehmen.

RC

Der Mikroprozessor führt einen Rücksprung aus, wenn das Carry-Bit ein L-Signal aufweist. Dieses H-Signal tritt auf, wenn ein Übertrag im Akkumulator entstanden ist. Das Hauptprogramm wird auf der festgehaltenen Adresse im Kellerspeicher fortgesetzt.

RNC

Tritt in dem Unterprogramm kein Übertrag auf, d.h. das Carry-Bit hat ein L-Signal, so verläßt der Mikroprozessor sein Unterprogramm und setzt sein Hauptprogramm auf der letzten Adresse vor seinem Unterprogramm-Sprung fort.

RZ

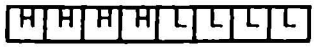
Der Mikroprozessor verläßt sein Unterprogramm, wenn das Ergebnis im Akkumulator gleich Null ist. Dadurch hat das Zero-Bit ein H-Signal und der Mikroprozessor springt auf die im Kellerspeicher gestapelte Adresse des Hauptprogramms zurück.

RNZ

Ist das Ergebnis des Akkumulators nicht gleich Null, so verläßt der Mikroprozessor sein Unterprogramm und begibt sich auf die im Stapel befindliche letzte Adresse seines Hauptprogrammes. Das Zero-Bit hat ein L-Signal, wenn der Inhalt des Akkumulators nicht gleich Null ist.

RM

Hat das Sign-Bit ein H-Signal, so hat der Inhalt des Akkumulators ein negatives Vorzeichen. Dadurch kann der Mikroprozessor sein Unterprogramm verlassen und führt sein Hauptprogramm auf der letzten Adresse vor dem Unterprogramm-Sprung fort.

RP 

Bei einem positiven Inhalt des Akkumulators kann der Mikroprozessor sein Unterprogramm verlassen. Das Sign-Bit hat ein L-Signal und der Mikroprozessor erhält einen Rücksprung-Befehl. Der Mikroprozessor führt sein Programm auf der letzten Adresse vor dem Unterprogramm-Sprung weiter.

RPE

Hat der Mikroprozessor in seinem Akkumulator eine gerade Parität, so wird ein Rücksprung ausgeführt, da das Parity-Bit ein H-Signal hat. Das Hauptprogramm wird auf der gestapelten Adresse in dem Kellerspeicher fortgesetzt.

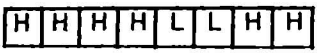
RPO

Tritt in dem Akkumulator eine ungerade Parität auf, d.h. das Parity-Bit hat ein L-Signal, so setzt der Mikroprozessor sein Hauptprogramm auf der gestapelten Adresse seines Kellerspeichers fort. Es handelt sich hierbei um einen bedingten Sprung.

Programmunterbrechungen.

EI

Durch diesen Befehl wird das INTE-Flip-Flop des Mikroprozessors gesetzt. Dadurch wird dem Mikroprozessor angezeigt, daß eine Unterbrechung (INTERRUPT) erkannt und berücksichtigt werden soll.

DI 

Mit diesem Befehl wird das INTE-Flip-Flop zurückgesetzt. Kommt nun ein Befehl für eine Programmunterbrechung, dann ignoriert der Mikroprozessor diesen Befehl.

Sonstige Befehle:

HLT

L	H	H	H	L	H	H	L

Tritt ein Halt-Befehl auf, wird der Befehlszähler auf die Adresse des nächstfolgenden Befehls erhöht. Der Mikroprozessor stoppt und wartet bis auf den nächsten INTERRUPT. Hat das INTE-Flip-Flop ein L-Signal, d.h. das INTERRUPT-System wurde ausgeschaltet, so muß der Mikroprozessor für einen weiteren Betrieb ausgeschaltet und anschließend wieder neu gestartet werden.

NOP

L	L	L	L	L	L	L	L

Bei dem Leer-Befehl wird keine Operation ausgeführt. Das Programm wird auf dem nächstfolgenden Befehl weitergeführt.

2.2. Taktgenerator 8224

In dem Bild 2.1 wird der 8080 von einem Taktgenerator angesteuert. Der Taktgenerator liefert für den 8080 die beiden Phasentakte $\emptyset_1$ und $\emptyset_2$.
In Bild 2.2.1 ist die Innenschaltung und in Bild 2.2.2 die Anschlußbelegung des Taktgenerators gezeigt. Für den Betrieb des Bausteines sind zwei Betriebsspannungen erforderlich, +5V für die TTL-Technik und +12V für die beiden direkten Phasentakt-Ausgänge $\emptyset_1$ und $\emptyset_2$.

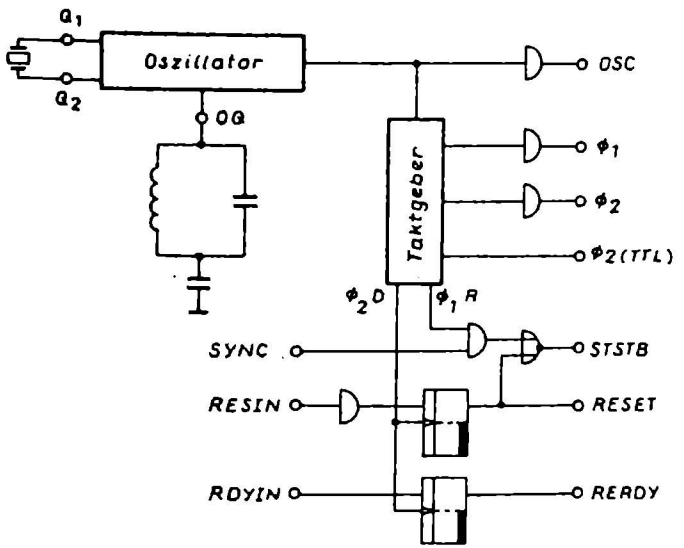


Bild 2.2.1 Innenschaltung des System-Steuerungs-Bausteines 8228 für das Mikroprozessor-System 8080

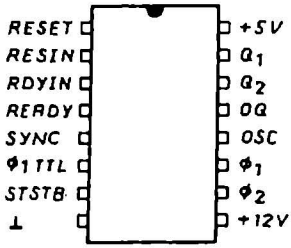


Bild 2.2.2 Anschlußbelegung des Taktgenerators 8224 (Draufsicht)

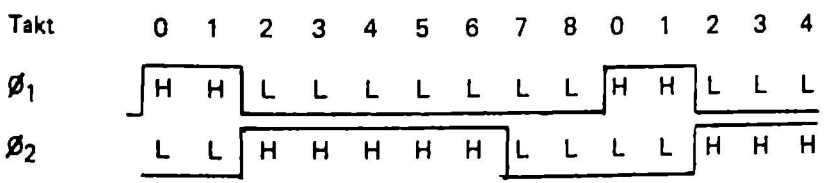
Über den RESIN-Eingang wird das interne D-Flip-Flop zurückgesetzt, wenn an dem Eingang ein L-Signal anliegt. An dem RESIN-Eingang befindet sich ein nicht-invertierender Schmitt-Trigger. Der Eingang kann damit an ein externes RC-Glied angeschlossen werden. Schaltet man die Betriebsspannung ein, so wird der Kondensator des RC-Gliedes langsam aufgeladen. Erreicht er eine bestimmte Schwellwert-Spannung, so erzeugt der nichtinvertierende Schmitt-Trigger-Ausgang eine sehr schnelle L-H-Flanke an dem D-Eingang des Rücksetz-Flip-Flops. Mit der nächsten L-H-Flanke des $\emptyset_2$-Taktes setzt sich das Flip-Flop und hat dann an seinem Q-Ausgang ein H-Signal. Der RESET-Ausgang hat ein H-Signal, aber der STSTB-Ausgang ein L-Signal.
Durch einen Schalter gegen Masse können wird den Kondensator des externen RC-Gliedes einfach löschen. Der Kondensator entlädt sich und der Schwellwert der RESIN-Eingangsspannung sinkt. Ab einer bestimmten Schwelle schaltet der Schmitt-Trigger um. An dem D-Eingang des D-Flip-Flops entsteht eine schnelle H-L-Impulsflanke. Bei dem nächsten L-H-Sprung des $\emptyset_2$-Taktes kippt das Rücksetz-Flip-Flop zurück und der RESET-Ausgang hat ein L-Signal. Durch die NOR-Gatter-Bedingung geht der STSTB-Ausgang auf H-Signal, wenn an dem SYNC-Eingang ein L-Signal anliegt. Der RESET-Ausgang ist der Rücksetz-Ausgang für das Mikroporozessor-System.
Der STSTB-Ausgang erzeugt uns für das Mikroprozessor-System ein Zustand-Übernahme-Signal. Der Mikroprozessor 8080 gibt zu Beginn eines Operationszyklus eine Zustandsinformation an den Daten-BUS ab. Damit kann kontrolliert werden, welchen Stand der Mikroprozessor hat. Der Mikroprozessor gibt auf den SYNC-Eingang ein H-Signal ab. Dieser Eingang ist über ein UND-Gatter mit dem internen $\emptyset_1$-Takt verbunden. Haben beide ein H-Signal, ist die UND-Bedingung erfüllt. An dem NOR-Gatter-Ausgang haben wir ein L-Signal. Dieses L-Signal ändert sich, wenn der interne $\emptyset_1$-Takt wieder auf L-Signal geht. Die UND-Bedingung ist nicht mehr erfüllt und wir haben an dem STSTB-Ausgang ein H-Signal.
Da der interne $\emptyset_1$-Takt kontinuierlich arbeitet, stellt das UND-Gatter eine Impulsweiche dar. Der STSTB-Ausgang gibt eine Taktimpulsfolge ab, wenn der SYNC-Eingang ein H-Signal hat und der RESET-Ausgang ein L-Signal aufweist. Hat dieser RESET-Ausgang ein H-Signal, kann die NOR-Gatter-Bedingung nicht erfüllt werden. Erst beim Rücksetzen des Rücksetz-Flip-Flops kann die NOR-Gatter-Bedingung wieder richtig arbeiten. Der STSTB-Ausgang ist direkt mit dem Steuer-Baustein 8228 verbunden.

Mit dem RDYIN-Eingang kann mit einem H-Signal das Bereit-Flip-Flop gesetzt werden, wenn der interne $\emptyset_2$-Takt eine positive L-H-Flanke abgibt. Das Flip-Flop setzt sich und der READY-Ausgang hat ein H-Signal.
Das Ausgangssignal von dem READY-Ausgang ist mit dem READY-Eingang des Mikroprozessors 8080 verbunden. Der 8080 benötigt für einen Zeitablauf genaue Signale, vor allen Dingen für das Vorbereiten und Halten. Durch den RDYIN-Eingang ergibt sich zusammen mit dem $\emptyset_2$-Takt eine synchrone Warteanforderung.

Erfolgt das Setzen des Bereit-Flip-Flops durch den $\emptyset_2$-Takt, kann der Baustein ein synchronisiertes Bereit-Signal direkt den 8080 ansteuern.
In dem Baustein 8224 befindet sich ein Oszillator. Durch einen externen Quarz bestimmen wir die Oszillator-Frequenz. Der Quarz kann direkt an die beiden Anschlüsse Q_1 und Q_2 angeschlossen werden. Hat der Quarz eine Frequenz über 10MHz, so soll unbedingt ein 10pF Kondensator mit dem Quarz in Reihe geschaltet werden.
Wenn wir einen Oberwellen-Quarz zur Ansteuerung verwenden, muß an dem OQ-Eingang ein kleiner Filter angeschlossen werden. Ein Oberwellen-Quarz hat den Vorteil, daß die Leistungsaufnahme sehr gering ist. Das Filter soll aus einer Spule und einem Kondensator bestehen. Die einzelnen Bauteile können aus der Formel

$$f = \frac{1}{2\pi\sqrt{L.C}}$$

ausgerechnet werden. Der OSC-Ausgang ist der direkte Oszillator-Ausgang. Diesen Ausgang können wir für meßtechnische Anwendungen verwenden.
Der Taktgeber des Bausteines 8224 ist ein 4-Bit-Synchron-Zähler. Er teilt die Oszillatorfrequenz im Verhältnis 1:9 herunter. Zwei Taktteile hat der $\emptyset_1$-Takt ein H-Signal und fünf Taktteile hat der $\emptyset_2$-Takt. Zwei Taktteile ergeben dann eine Impulspause. Es ergibt sich folgendes Impulsdiagramm:



Der 4-Bit-Synchron-Zähler teilt die Frequenz herunter und der Decodierer decodiert die Ausgänge des Zählers nach diesem Impulsdiagramm.
Die Ausgänge des Decoders erzeugen die beiden Taktphasen $\emptyset_1$ und $\emptyset_2$ für den Mikroprozessor 8080. Die Ausgangsspannung beträgt +12V. Für zusätzliche Steuerungszwecke hat der 8224 nach einem $\emptyset_2$ (TTL)-Ausgang. Der Ausgang liefert Signale für die TTL-Technik. Mit dem Signal können wir einem anfragenden Baustein einen BUS zuteilen, wenn der 8080 eine HOLD-Quittung ausgibt.

2.3 System-Steuerungs-Baustein 8228

In dem Bild 2.1 befindet sich der System-Steuerungs-Baustein zwischen dem Mikroprozessor 8080 und den BUS–Systemen.
In Bild 2.3.1 ist die Innenschaltung und in Bild 2.3.2 die Anschlußbelegung gezeigt. Der Baustein besteht aus drei Gruppen:

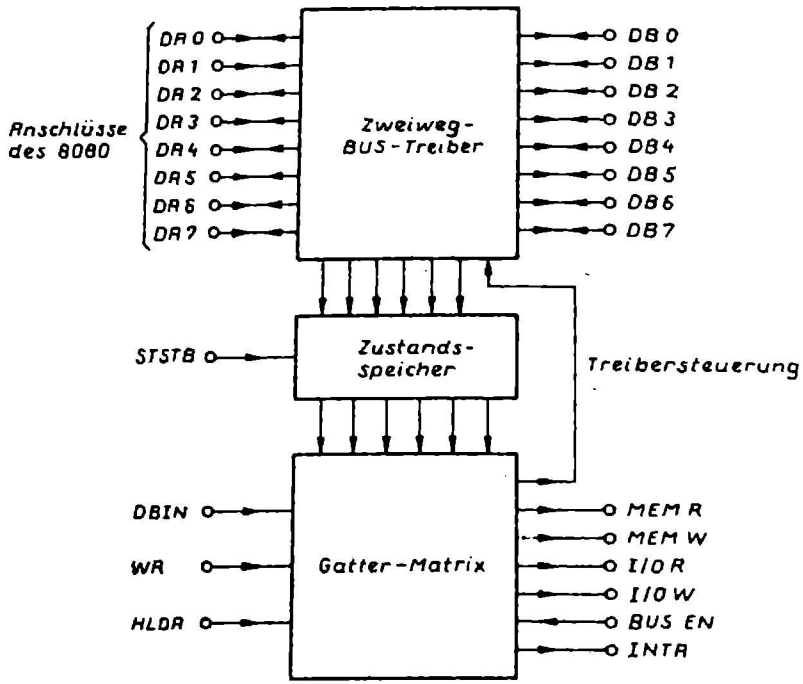


Bild 2.3.1 Innenschaltung des Systems-Steuerungs Bausteines 8228 für das Mikroprozessor-System 8080

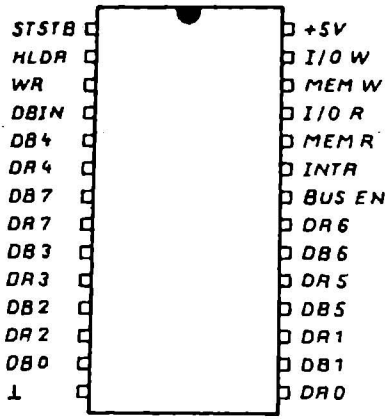


Bild 2.3.2 Anschlußbelegung des System-Steuerungs-Bausteines 8228 (Draufsicht)

a) Der Zweiweg–BUS-Treiber kann Daten von links nach rechts und umgekehrt transportieren. Durch diesen Treiber können wir den 8080 von dem BUS-System trennen. Die D_A-Ein/Ausgänge sind direkt für den Mikroprozessor 8080 zu verwenden. Diese Ein/Ausgänge erzeugen eine Eingangsspannung von über 3,3 V bei einem maximalen Strom von 2mA. Damit kann der 8080 angesteuert werden.
Die D_B-Ein/Ausgänge können einen Strom bis zu 40mA liefern.

b) Beginnt der Mikroprozessor einen neuen Befehlszyklus, gibt er eine Zustands-Information ab. Diese Zustands-Information wird in dem Zustandsspeicher gespeichert, wenn an dem STSTB-Eingang ein L-Signal anliegt. Damit wird der Zustand des Daten-BUS direkt gespeichert und an den Gatter-Matrix weitergegeben.

c) Die gespeicherten Daten des 8080 werden in der Gatter-Matrix zu Steuersignalen umgewandelt. Insgesamt erzeugt die Gatter-Matrix folgende Befehle:

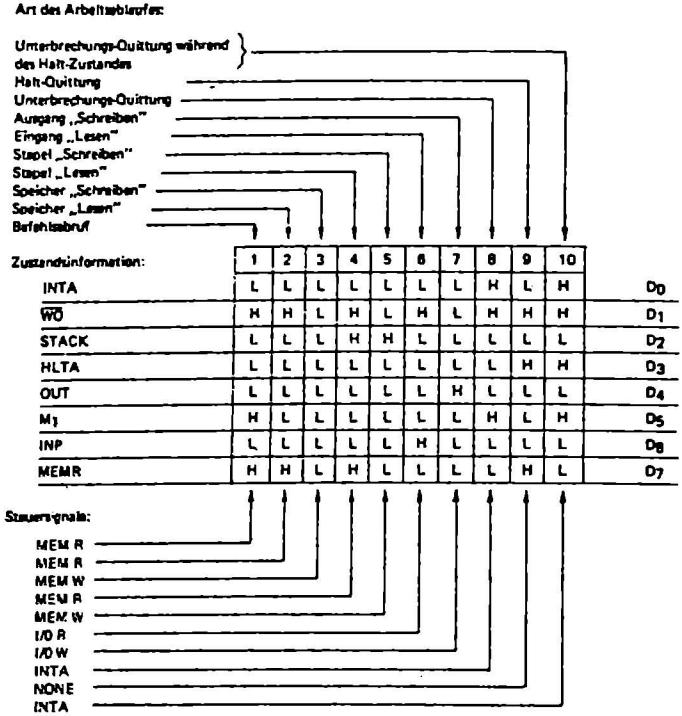
MEM R:	Speicher lesen oder "memory read"
MEM W:	Speicher einschreiben oder "memory write"
I/O R:	Eingang/Ausgang lesen oder "input/output read"
I/O W:	Eingang/Ausgang schreiben oder "input/output write";
INTA:	Unterbrechungs-Quittung oder "interrupt acknowledge";

Mit diesen fünf Befehlen erhalten wir den Steuer-BUS von Bild 2.1. Mit dem Steuer-BUS können wir die einzelnen Speicher ansteuern.

Die drei Steuersignale für den Lese-Betrieb (MEM R, I/O R und INTA) werden durch logische Verknüpfung mit dem zugehörigen Zustands-Bit und den DBIN-Eingang (data bus input) des 8080 erzeugt.
Die zwei Steuersignale für den Schreib-Betrieb (MEM W, I/O W) werden durch die logische Verknüpfung der Zustands-Bits und des WR-Einganges des 8080 erzeugt.
Die Steuersignale des 8228 können direkt mit den Steuer-Eingängen der RAMs und ROMs verbunden werden. Die Verbindungen laufen dabei über den Steuer-BUS.
Mit dem INTA-Ausgang kann die Programmunterbrechung des 8080 signalisiert werden. Dies ist aber nur für Eingabe- und Ausgabe-Einheiten interessant. Wird dieser Eingang nicht benötigt, so ist er auf +12V zu legen, aber über einen 1kΩ Widerstand. Dadurch gibt der 8228 zur geeigneten Zeit einen RST-Befehl aus.

Bei der Verwendung eines CALL-Befehles als Unterbrechungs-Befehl, erzeugt der 8228 einen INTA-Impuls für jede der drei Befehls-Bytes.
Der BUS EN-Eingang bei der Gatter-Matrix ist ein asynchroner Eingang, der die Daten-BUS-Ausgänge und die Steuersignal-Ausgänge in einen hochohmigen Zustand versetzt, wenn der BUS-EN-Eingang auf H-Signal liegt. Hat der Eingang dagegen ein L-Signal, arbeiten die Daten-Buffer und die Steuersignale normal.

Für den Baustein 8228 ergeben sich folgende Zustands-Bezeichnungen:



D_0 bis D_7:	Daten-BUS für die Seite des Mikroprozessors
I/O R:	Lesen der Ein/Ausgabe
I/O W:	Schreiben der Ein/Ausgabe
MEM R:	Speicher lesen
MEM W:	Speicher schreiben
DBIN:	Daten-BUS-Steuerung vom Mikroprozessor aus
INTA:	Unterbrechungs-Quittierung
HLDA:	Haltesteuerung vom Mikroprozessor aus
WR:	Schreiben vom Mikroprozessor aus
BUS EN:	BUS-Freigabe-Eingang
STSTB:	Zustandsübernahme vom Taktgenerator 8224 aus

2.4 Schnittstellen

in Bild 2.1 befindet sich zwischen dem Mikroprozessor und dem Adressen-BUS eine Adressen-Schnittstelle. Für diese Schnittstelle können folgende Bausteine verwendet werden.

8212: 8-Bit-Ein/Ausgabe-Baustein
8205: Schneller 1- aus 8-Binär-Decoder
8216: 4-Bit-parallele Zweiweg-BUS-Treiber (nichtinvertierend)
8226: 4-Bit-parallele Zweiweg-BUS-Treiber (invertierend)

Der Baustein 8212 ist ein 8-Bit-paralleles Daten-Register mit Buffer. In Bild 2.4.1 ist die Innenschaltung und in Bild 2.4.2 die Anschlußbelegung gezeigt.

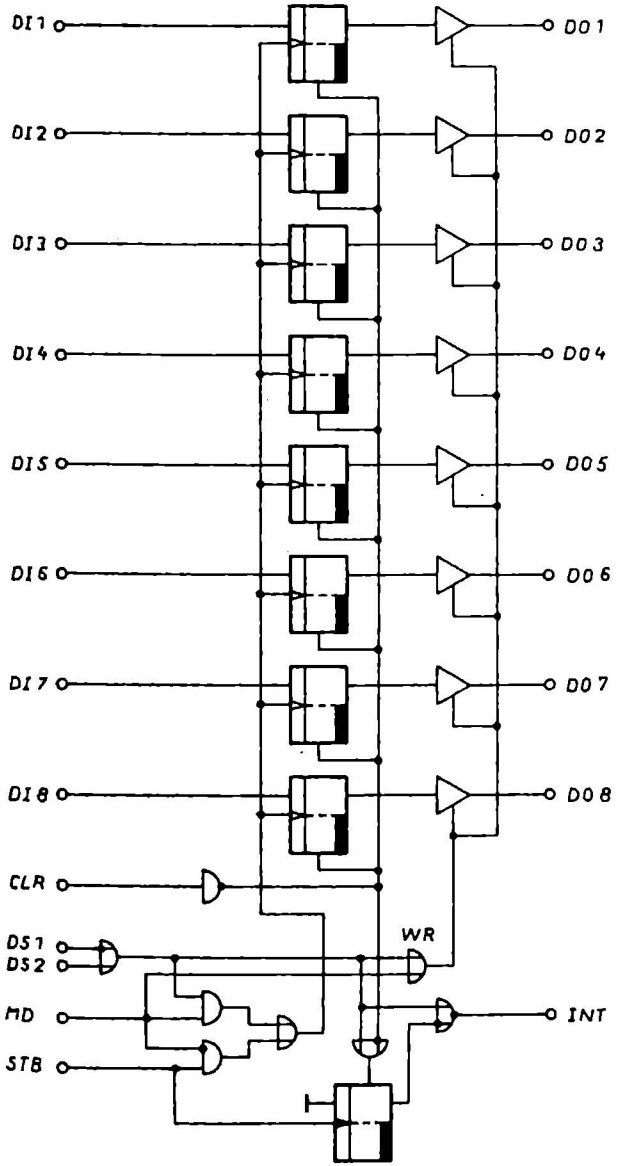


Bild 2.4.1 Innenschaltung des 8-BitEin/Ausgabe-Bausteines 8212

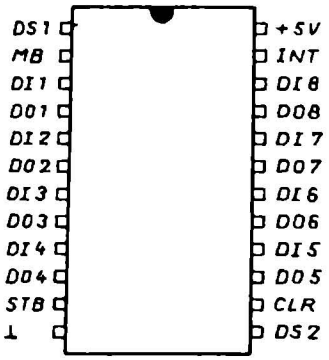


Bild 2.4.2 Anschlußbelegung des 8-Bit-Ein/Ausgabe-Bausteines 8212 (Draufsicht)

Über die acht Daten-Eingänge DI_1 bis DI_8 liegen die Eingangsinformationen an. Diese Eingangsinformationen erhält der Baustein 8212 von dem Mikroprozessor-Baustein. Die Eingangsinformationen sind die Adressen-Ausgänge des 8080. In den Flip-Flops werden die Adressen zwischengespeichert. Die Ausgabe der Adressen erfolgt über die Ausgangs-Buffer, die einen Tri-State-Ausgang haben. Die DO_1 bis DO_8-Ausgänge steuern direkt den Adressen-BUS an.
Der Eingangsstrom für einen Eingang beträgt 0,25mA, was einem "fan in" von n=0,15 entspricht. Der Ausgangsstrom zieht gegen L-Signal einen Strom von 15mA ("fan out" von n=10). Die Speicher-Flip-Flops können asynchron durch den CLR-Eingang gelöscht werden.

Der Baustein 8212 ist für mehrere Betriebsanwendungen ausgelegt. Der 8212 hat ein internes Bedienungs-Flip-Flop zur Erzeugung und Steuerung von Unterbrechungen für den Mikroprozessor. Der Baustein kann als Signalspeicher, als gesteuerter Buffer oder Multiplexer verwendet werden.
Die acht internen Speicher-Flip-Flops sind D-Flip-Flops. An den D-Eingängen liegen die Eingangsinformationen an. Mit einem H-Signal an den D-Eingängen setzen sich die Flip-Flops und mit einem L-Signal kippen sie zurück, wenn das Taktsignal von H nach L übergeht. Bei einem L-Signal an dem Takteingang der Flip-Flops bleibt der Signalzustand gespeichert.
Liegt an den Takt-Eingängen der Flip-Flops ein H-Signal an, schalten die Flip-Flops jede Änderung der Signalzustände an den D-Eingängen sofort durch. Die Speicherung der Information erfolgt bei einem H-L-Sprung des Taktes. Soll der Inhalt der Flip-Flops gelöscht werden, so geben wir auf den CLR-Eingang ein L-Signal. Damit werden die Flip-Flops zurückgekippt und die Ausgänge haben ein L-Signal. Bei einem L-Signal an den Takt-Eingängen bleibt diese Rückstellung aus.
Die Ausgänge der acht Flip-Flops sind über nichtinvertierende Ausgangs-Buffer verschaltet. Die Ausgänge können drei Ausgangs-Zustände annehmen, d.h. wir haben Tri-State-Ausgänge. Die Ausgangs-Buffer haben eine gemeinsame Steuerleitung. Die Steuerung der Ausgangs-Buffer wird über das WR-ODER-Gatter vorgenommen. Hat der Ausgang ein Signal für die Ausgangs-Freigabe, so werden die Ausgangsinformationen der D-Flip-Flops direkt auf den Ausgang geschaltet. Bei einem Sperr-Signal sind die Ausgänge des Buffers hochohmig. Dadurch können wir die Ausgänge des 8212 direkt auf den Adressen-BUS schalten.

Der Baustein 8212 hat eine interne Steuer-Logik. Hierfür sind die Signale an den Eingängen DS_1, DS_2, MD und STB zuständig. Sie steuern durch ihre Signal-Zustände die Baustein-Auswahl, die Daten-Signalspeicherung, die Ausgangs-Buffer (Tri-State-Steuerung) und das Bedienungsanforderungs-Flip-Flop.
Die beiden DS-Eingänge dienen zur selektiven Eingangssteuerung. Hat der DS_1-Eingang ein L-Signal und der DS_2 ein H-Signal, so ist der Baustein selektiert, d.h. er kann arbeiten. Die beiden Eingänge sind über ein UND-Gatter verknüpft, wobei der DS_1-Eingang über ein NICHT-Gatter an dem UND-Gatter anliegt. Im selektierten Zustand werden die Ausgangs-Buffer freigegeben, während das Bedienungsanforderungs-Flip-Flop asynchron gesetzt wird. Das Flip-Flop kann direkt durch die beiden Eingänge gesetzt werden, wenn die UND-Gatter-Bedingung erfüllt ist. Der Ausgang des Flip-Flops hat damit ein L-Signal und der INT-Ausgang des Bausteines geht auf L-Signal.

Mit dem MD-Eingang wird die Betriebsart bei dem 8212 bestimmt. Das Eingangssignal dient zur Steuerung des Zustandes des Ausgangs-Buffers und bestimmt die Quelle des Taktsignales für die acht D-Flip-Flops. Hat der MD-Eingang ein H-Signal, sind die Ausgangs-Buffer freigegeben, d.h. die Q-Ausgänge der acht D-Flip-Flops werden direkt auf die Ausgänge geschaltet. Das Takteingangssignal für die acht D-Flip-Flops stammen bei einem H-Signal von den beiden Selektions-Eingängen DS_1 und DS_2.

Hat der MD-Eingang ein L-Signal, so wird der Zustand des Ausgangs-Buffer von den beiden Selektions-Eingängen DS_1 und DS_2 bestimmt. Das Taktsignal für die acht D-Flip-Flops erhalten wir durch den STB-Eingang (Übernahme).

Der STB-Eingang ist der Strobe-Übernahme-Eingang. Der Eingang dient als Takt-Eingang für die acht D-Flip-Flops, vorausgesetzt, der MD-Eingang hat ein L-Signal. Das Bedienungsanforderungs-Flip-Flop wird mit einer negativen Flanke des STB-Eingangs getriggert.

Das Bedienungsanforderungs-Flip-Flop wird zur Erzeugung und Steuerung von INTERRUPTs in einem Mikroprozessor-System benötigt. Das Flip-Flop wird asynchron gesetzt durch den CLR-Eingang. Mit einem L-Signal wird das Flip-Flop an CLR gesetzt. Ist das Flip-Flop gesetzt, befindet es sich im Nichtunterbrechungs-Zustand. Hier kann dann kein INTERRUPT durchgeführt werden.

Der Q-Ausgang des Bedienungsanforderungs-Flip-Flops ist an dem invertierenden NOR-Gatter-Eingang angeschlossen. Der andere Eingang für das NOR-Gatter ist direkt mit dem UND-Gatter für die Baustein-Selektion verbunden. Der Ausgang des NOR-Gatters hat im L-Signalzustand einen Unterbrechungs-Zustand und wird mit den Eingängen der Bausteine zur Prioritäts-Signalerzeugung angeschlossen.

Die einfachste Anwendung des 8212 ist eine Schnitt-Stelle zwischen dem Mikroprozessor und dem Adressen-BUS. Wenn wir den MD-Eingang auf L-Signal und den STB-Eingang auf H-Signal legen, arbeitet der 8212 als durchgeschaltetes Gatter. Die Ausgangs-Buffer werden in diesem Fall durch die DS-Eingänge freigegeben. Diese Schnitt-Stelle dient als Entkoppelung zwischen dem Mikroprozessor 8080 und dem Adressen BUS.

Ein weiterer Anwendungsfall ist in Bild 2.4.3 gezeigt. Hier arbeitet der 8121 als Zustands-Signal-Speicher. Der 8212 übernimmt die Funktion des Zustands-Signal-Speichers für den Mikroprozessor 8080.

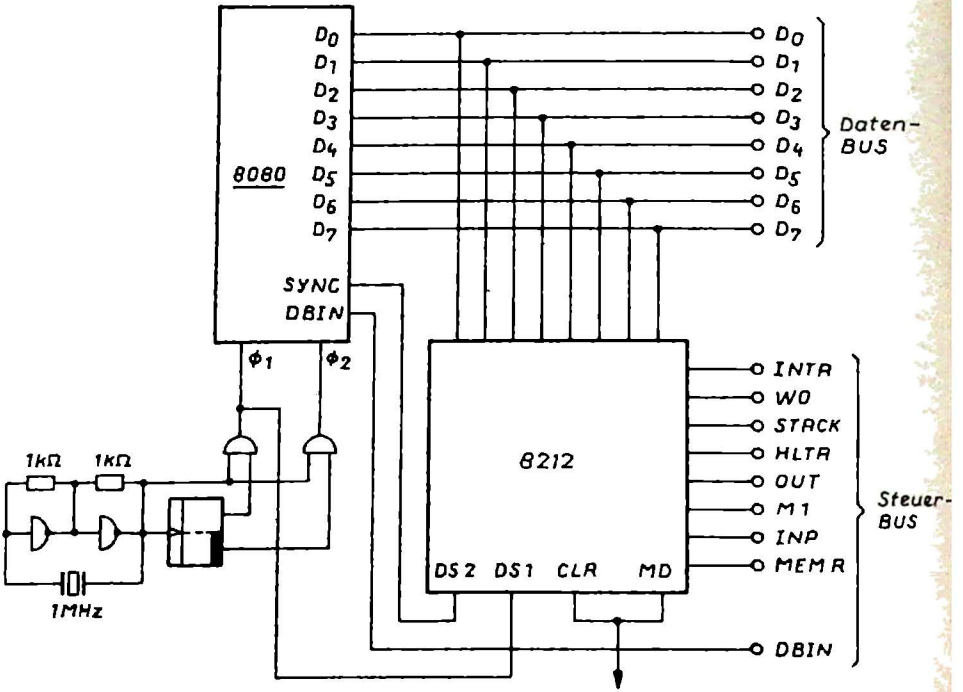


Bild 2.4.3 Der Baustein 8212 arbeitet als Zustands-Signal-Speicher

Der MD- und der CLR-Eingang liegen an L-Signal. Der Inhalt der acht internen Speicher-Flip-Flops ist dadurch gelöscht. Bei einer Selektion durch den $Ø_2$ TTL-Takt und durch den SYNC-Ausgang des 8080 findet eine Selektion des Bausteines statt und die Q-Ausgänge der Flip-Flops werden auf den Ausgang geschaltet, aber die Flip-Flops sind gelöscht.
Die internen Speicher-Flip-Flops können gesetzt werden, wenn an dem MD- und an dem CLR-Eingang ein H-Signal anliegt. Durch die Selektion der DS-Eingänge erfolgt die Ansteuerung. Die Daten der DI-Eingänge werden erst dann übernommen, wenn der DS_1-Eingang ein H-Signal erhält. Hiermit ist die Selektion beendet und an den Takt-Eingängen der Speicher-Flip-Flops ist ein H-L-Sprung entstanden. Die Informationen liegen an den Ausgängen an, da der MD-Eingang ein H-Signal hat.

INTA:

Mit diesem Signal wird ein Quittungssignal für eine Unterbrechungs-Anforderung ausgesendet. Der Zustand wird auch verwendet, um einen RESTART-Befehl auf den Daten-BUS zu leiten, wenn der DBIN-Ausgang (Daten-BUS-Steuerung) aktiv ist.

WO:

Der Zustand zeigt an, daß eine Operation im gegenwärtigen Operationszyklus entweder eine Speicher-Schreib oder eine Ausgabe-Operation ist. Die Speicher-Schreib-Operation wird auf dem Daten-BUS mit einem H-Signal auf dem Daten-Bit D_1 angezeigt, d.h. auf der zweiten Bitstelle von rechts. Hat das Daten-Bit D_1 ein L-Signal, wird eine Ausgabe-Operation signalisiert.

STACK:

Mit diesem Zustand wird angezeigt, daß auf dem Adressen-BUS die Adresse des Stapelzeigers liegt.

HLTA:

Der Ausgang zeigt das Quittungs-Signal für den HALT-Befehl.

OUT:

Der Ausgang zeigt uns an, daß sich die Adresse eines Ausgabekanals auf dem Adressen-BUS befindet und daß die Ausgabe-Daten auf dem Daten-BUS liegen, wenn der WR-Eingang des Mikroprozessors 8080 aktiv wird.

M_1

Der Ausgang zeigt an, daß sich der Mikroprozessor 8080 in einem Abruf-Zyklus zum Einholen des ersten Befehls-Byte befindet.

INP:

Dieser Zustand zeigt an, daß sich auf dem Adressen-BUS eine Adresse von einer Eingabeeinheit befindet, wenn DBIN aktiv wird. Auf dem Daten-BUS befinden sich dann die Daten der Eingabeeinheit. Wir adressieren eine Eingabeeinheit und schalten damit die Eingangs-Daten auf den Daten-BUS.

MEMR.

Der Ausgang nimmt einen Zustand ein, wenn der Daten-BUS die Schreib-Lese-Daten annimmt.

In Bild 2.4.4 ist der Aufbau eines sehr einfachen Mikroprozessor-Systems gezeigt. Das System besteht im wesentlichen nur aus dem Mikroprozessor 8080, einem einfachen Taktgenerator, einem Festwertspeicher (ROM) für die Befehle, einem Schreib-Lese-Speicher für die Datenzwischenspeicherung, einem Zustandsspeicher und zwei Schnittstellen für die Eingabe und die Ausgabe der Informationen.

Der Adressen-BUS steuert direkt das ROM und das RAM an. Das ROM besteht in diesem Fall aus dem Baustein 8316 (löschbares und elektrisch neu programmierbares PROM). Wir können in diesem PROM insgesamt 2048 Befehle speichern. Für das RAM verwenden wir die statischen Schreib-Lese-Speicher vom Typ 8101 (256 x 4-Bit). Wir schalten zwei RAMs parallel und erhalten somit eine 8-Bit-Struktur mit insgesamt 256 Speicherplätzen.
Das ROM liegt über den Baustein 8212 an. Dies hat den Vorteil, daß wir die beiden Speicher, also das ROM und das RAM gleichzeitig adressieren können, aber der ROM-Inhalt kann nur ausgelesen werden, wenn der DS_1-Eingang an L-Signal und der DS_2-Eingang an H-Signal liegt. Hat der DBIN-Ausgang (Daten-BUS-Steuerung) ein H-Signal und der MEMR-Ausgang ein H-Signal, nimmt der Daten-BUS die ausgelesenen Daten des ROMs an. Wir können damit einen Befehl in den Mikropro-

zessor oder eine Konstante einschreiben. Zuvor muß aber der Zustands-Speicher, der ebenfalls aus dem Baustein 8212 besteht, ein MEMR-Signal abgegeben haben.

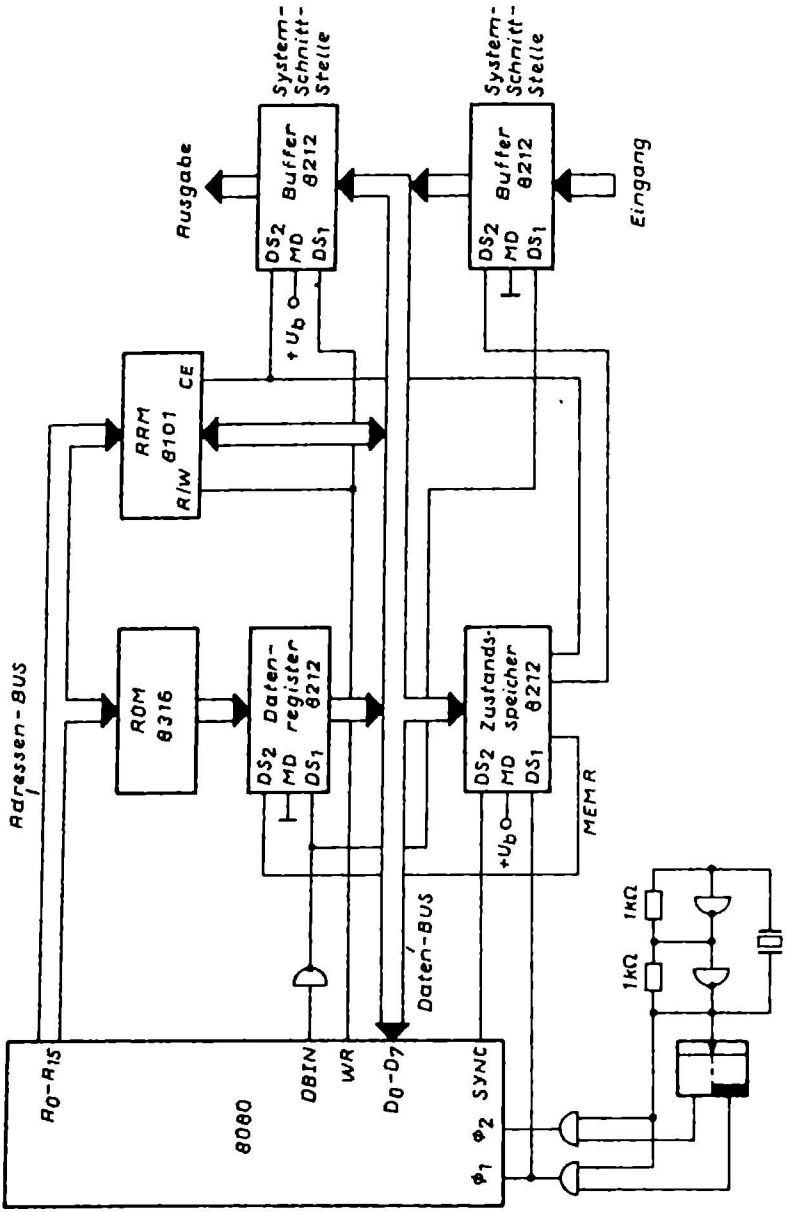


Bild 2.4.4 Aufbau eines einfachen Mikroprozessor-Systems mit Mikroprozessor, Taktgenerator, Befehlsspeicher (ROM), Datenspeicher (RAM) und einer Ausgabe- und einer Eingabe-Einheit

Das Einschreiben oder Auslesen von Informationen aus dem RAM ist von dem Zustands-Speicher abhängig. Der Mikroprozessor gibt auf seinem WR-Ausgang (Schreiben) ein L-Signal ab und in den Schreib-Lese-Speicher kann eingeschrieben werden, wenn der CE-Eingang ein L-Signal von dem Zustands-Speicher erhält. In

das RAM können Daten zur Zwischenspeicherung eingeschrieben werden. Während dieser Zeit ist die System-Ausgabe gesperrt, da an dem DS_2-Eingang ein L-Signal anliegt.
Die Daten des Mikroprozessor-Systems können die Daten über den Ausgabe-Baustein (Parallel-Schnitt-Stelle) verlassen, wenn der WR-Ausgang des Mikroprozessors ein L-Signal und der Zustands-Speicher ein H-Signal auf den DS_2-Eingang gibt.
Die Daten aus dem Schreib-Lese-Speicher können in den Mikroprozessor eingeschrieben werden, wenn der R/W-Eingang des RAMs ein H-Signal von dem W/R-Ausgang des Mikroprozessors und ein L-Signal von dem Zustands-Speicher erhält. In diesem Fall gibt das RAM einen adressierten Speicherplatz zum Auslesen frei.
Der MD-Eingang für die Betriebsart liegt bei der Ausgabe-Einheit immer auf H-Signal. Daher sind die Daten-Ausgänge der Flip-Flops für die nachgeschaltete Schaltung kontinuierlich erreichbar. Die Eingangs-Daten werden gespeichert, wenn sich das Signal an dem WR-Ausgang des Mikroprozessors von L nach H ändert. Die so gespeicherten Informationen des Mikroprozessor-Systems bleiben bis zur nächsten Ausgangs-Informationsänderung erhalten.
Benötigt der Mikroprozessor neue externe Daten, so gibt er auf die Eingabe-Schnitt-Stelle einen entsprechenden Befehl ab. Der MD-Eingang des Bausteines 8212 liegt immer auf L-Signal. Daher sind im Ruhezustand die Ausgangs-Buffer des Bausteines gesperrt, d.h. sie sind hochohmig. Die Selektion des Bausteines erfolgt durch das Mikroprozessor-System. Der Zustands-Speicher und der DBIN-Ausgang des Mikroprozessors 8080 geben ein H-Signal ab. Damit werden die Eingangsinformationen auf den Daten-BUS geschaltet. Während dieses Einschreibens der Informationen kann sich aber der Zustand des Informationen ändern, da bei dieser Beschaltung des Bausteines die Informationen direkt durchgeschaltet werden.

Mit der Schaltung von Bild 2.4.4 kann ein Programm-Zyklus ablaufen. Zuerst adressiert der 8080 sein ROM und die ROM-Schnitt-Stelle 8212. Der Mikroprozessor holt sich seinen Befehl. Der Befehl wird im 8080 decodiert. Danach holt sich der Mikroprozessor einen Operanden. Der Operand befindet sich entweder im RAM oder am System-Eingang.
Nach dem Einholen des Operanden wird der Befehl ausgeführt. Der Operand, der verarbeitet worden ist, wird nun entweder im RAM zwischengespeichert oder über die System-Ausgabe ausgegeben.
Nach der Befehlsausführung erhöht der Befehlszähler des 8080 seinen Zählerstand und holt sich damit einen neuen Befehl. Danach läuft das Prozeßschema erneut ab.

Den Befehls-Ablauf bezeichnet man als Neumann-Zyklus:

1) Befehl holen
2) Befehl decodieren
3) Operanden holen
4) Befehl ausführen und Ergebnis ausgeben oder speichern
5) Stand des Befehlszählers erhöhen

Als Schnitt-Stelle in Bild 2.1 zwischen dem Mikroprozessor und dem Adressen-

BUS kann auch der Baustein 8205 verwendet werden. In Bild 2.4.5 ist die Schaltung und in Bild 2.4.6 die Anschlußschaltung gezeigt. Der Baustein 8205 ist ein schneller 1- aus 8-Binär-Decoder.

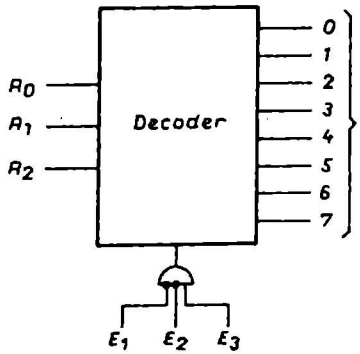


Bild 2.4.5 Innenschaltung des schnellen 1-aus-8-Decoder-Bausteines 8205 für das Mikroprozessor-System 8080

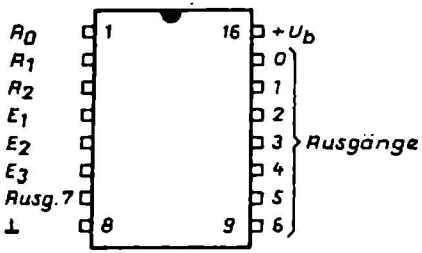


Bild 2.4.6 Anschlußbelegung des Bausteines 8205 (Draufsicht)

Der Baustein kann als Ein/Ausgabe oder als Speicherwahl-Schnitt Stelle verwendet werden. Hat der 8205 ein Freigabe-Signal, sind seine acht Ausgänge auf L-Signal. Auf diese Weise ist eine Zeile eines Speichersystems angewählt.

Durch die drei Baustein-Freigabe-Eingänge ist eine einfache Systemerweiterung möglich. Es ergibt sich folgende Funktionstabelle:

Adresse			Freigabe			Ausgänge							
A_2	A_1	A_0	E_1	E_2	E_3	0	1	2	3	4	5	6	7
L	L	L	L	L	H	H	L	L	L	L	L	L	L
L	L	H	L	L	H	L	H	L	L	L	L	L	L
L	H	L	L	L	H	L	L	H	L	L	L	L	L
L	H	H	L	L	H	L	L	L	H	L	L	L	L
H	L	L	L	L	H	L	L	L	L	H	L	L	L
H	L	H	L	L	H	L	L	L	L	L	H	L	L
H	H	L	L	L	H	L	L	L	L	L	L	H	L
H	H	H	L	L	H	L	L	L	L	L	L	L	H
X	X	X	L	L	L	H	H	H	H	H	H	H	H
X	X	X	H	L	L	H	H	H	H	H	H	H	H
X	X	X	L	H	L	H	H	H	H	H	H	H	H
X	X	X	H	H	L	H	H	H	H	H	H	H	H
X	X	X	H	L	H	H	H	H	H	H	H	H	H
X	X	X	L	H	H	H	H	H	H	H	H	H	H
X	X	X	H	H	H	H	H	H	H	H	H	H	H

Eine andere Möglichkeit in Bild 2.1 ist die Verwendung von Zweiweg-BUS-Treiber-Bausteinen für die Schnitt-Stelle.In Bild 2.4.7 ist die Innenschaltung des Zweiweg-BUS-Treiber-Bausteines 8216 gezeigt. Der Baustein arbeitet im nichtinvertierenden Betrieb, d.h. die Ausgangs-Daten sind mit den Eingangs-Daten phasenidentisch.

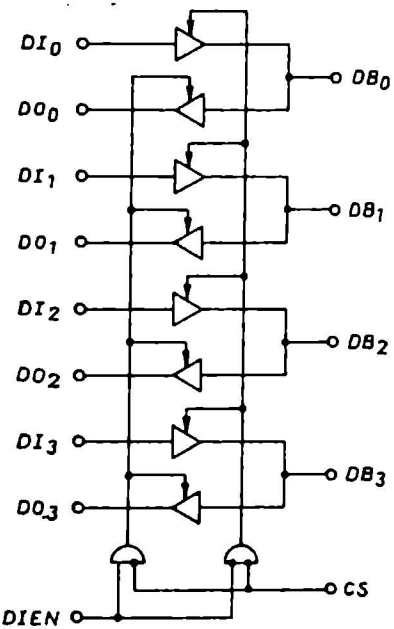


Bild 2.4.7 Innenschaltung des Zweiweg-BUS-Treiber-Bausteines 8216 für das Mikroprozessor-System 8080

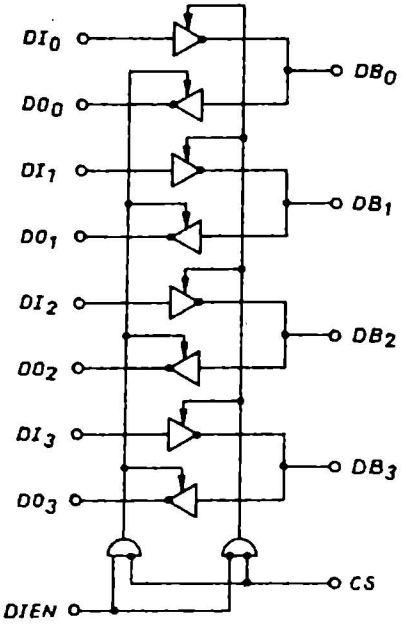


Bild 2.4.8 Innenschaltung des Zweiweg-BUS-Treiber-Bausteines 8226

Ein weiterer Zweiweg-BUS-Treiber ist der Baustein 8226. (Bild 2.4.8) Die hier verwendeten Ausgangs-Treiberstufen invertieren das Ausgangs-Signal um 180°. In der Mikroprozessoren-Technik verwenden wir aber meistens den Baustein 8216, da bei dem nichtinvertierenden BUS-Treiber die Signale bei der Ausgabe oder Eingabe von Daten immer invertiert anliegen müssen.

In Bild 2.4.9 ist das Anschlußbild gezeigt.

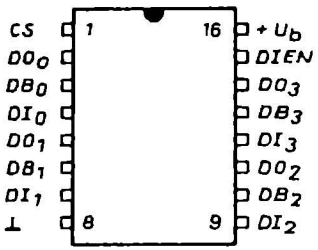


Bild 2.4.9 Anschlußbelegung des Bausteines 8216 und 8226 (Draufsicht)

Jede Einheit besteht in dem Baustein 8216 aus zwei Treiberstufen. Über den DI-Eingang liegen die Eingangs-Daten an. Die Eingangs-Daten stehen an dem DB-Ausgang zur Verfügung. Gleichzeitig kann der DB-Ausgang auch als Eingang verwendet werden. Der Ausgang hat daher die DO-Bezeichnung.
Jede Einheit des Bausteines hat zwei getrennte Buffer, die aber zusammengeschaltet werden können. Verbinden wird die DI-Anschlüsse mit den DO-Anschlüssen, so

haben wir einen Zweiweg-Buffer oder eine Zweiweg-Schnitt-Stelle. Dies können wir aber nur daher durchführen, da sämtliche Ausgangsstufen ein Tri-State-Verhalten haben.
Der Baustein 8216 und 8226 wird von den DIEN- und CS-Steuersignalen gesteuert. Der CS-Eingang wird für die Selektion des Bausteines verwendet. Hat der Eingang ein H-Signal, so befinden sich alle Treiber-Ausgänge im hochohmigen Zustand. Liegt der CS-Eingang dagegen an L-Signal, haben wir eine Selektion des Bausteines und die Ausgangsstufen der Treiber-Ausgänge können arbeiten, d.h. die Eingangsinformationen übertragen.

Die Datenflußrichtung wird dagegen mit dem DIEN-Eingang gesteuert. Hat der Eingang ein H-Signal, so erfolgt die Durchschaltung von den DB-Eingängen zu den DO-Ausgängen. Legen wir an den DIEN-Eingang ein L-Signal, fließen die Daten von den DI-Eingängen zu den DB-Ausgängen.

Der Baustein 8216 und 8226 kann als Schnitt-Stelle zwischen einem Speicher und einer Eingabe/Ausgabe-Einheit verwendet werden. In Bild 2.4.10 ist der Aufbau einer solchen Schnitt-Stelle gezeigt.
Über den Daten-BUS liegen die Daten an den beiden Daten-Buffern an. Für den Speicher kann entweder der 8216 oder der 8226 verwendet werden. Mit dem MEMR-Signal des Zustands-Speichers bestimmen wir den Schreib- oder Lesebetrieb in dem Speicher. Hat die MEMR-Leitung ein L-Signal, so können die Informationen des Daten-BUS auf die selektierte Speicheradresse eingeschrieben werden. Der Datenfluß findet von dem Daten-BUS zum Speicher statt. Hat die MEMR-Leitung dagegen ein H-Signal, fließen die gespeicherten Daten von dem Speicher in den Daten-BUS. Gesteuert wird der Buffer mit dem CS-Eingang.

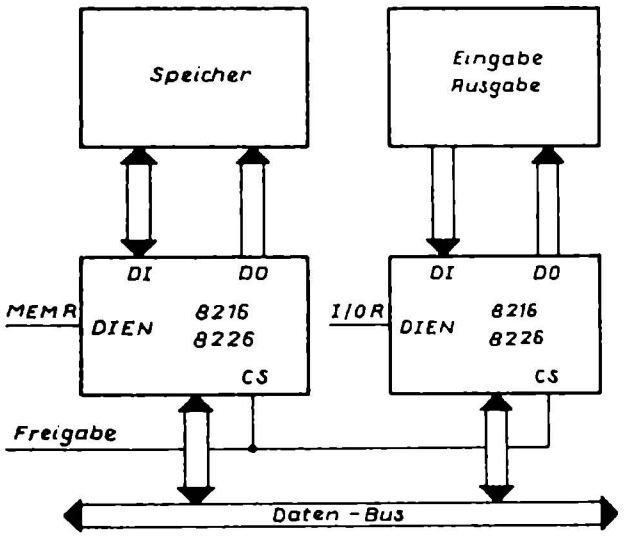


Bild 2.4.10 Schnittstelle mit einem Zweiweg-BUS-Treiber-Baustein 8216 oder 8226 zwischen einem Speicher und einer Eingabe/Ausgabe-Einheit

Bei der Eingabe/Ausgabe stellt der Baustein 8216 und 8226 eine Schnitt-Stelle zwischen den externen Baueinheiten und dem Mikroprozessor dar. Zweckmäßigerweise verwendet man hier den nichtinvertierenden Baustein. Mit dem I/O R-Signal des Zustands-Speichers bestimmen wir den Datenfluß aus dem Mikroprozessor-System oder den Datenfluß hinein. Hat die I/O R-Leitung ein L-Signal, so werden die Informationen des Daten-BUS auf den Ausgang gegeben,. d.h. die Mikroprozessor-System-Daten stehen uns zur Verfügung. Sollen Daten hineingeschrieben werden, so muß die I/O R-Leitung ein H-Signal haben. Die externen Daten gelangen so in das Mikroprozessor-System.

Die Treiber-Bausteine haben auch noch den Vorteil, daß wir längere Leitungen betreiben können. Ebenso können wir andere Peripherie-Geräte, wie Lochstreifenleser, Lochstreifenstanzer, einen Fernschreiber usw. anschließen.
In Bild 2.4.11 ist eine einfache Mikroprozessor-Schaltung gezeigt. Mittelpunkt ist hierbei wieder der 8080. Die Adressen-Ausgänge des 8080 steuern das ROM und das RAM an. Wichtig bei dieser Ansteuerung ist der Steuerungs-Betrieb. Von der Adresse A_{15} hängt es ab, ob sich der Mikroprozessor auf den Adressen der Speicher befindet, oder ob er sich in der Eingabe-oder Ausgabe-Einheit befindet.
Hat die Adressen-Leitung A_{15} ein L-Signal, so werden die ROM- und RAM-Einheiten angesteuert. Hat die Adressen-Leitung A_{15} dagegen ein H-Signal, sind die ROM- und RAM-Einheiten gesperrt, aber die Eingabe–und Ausgabe-Einheiten können arbeiten.
Die Daten-Anschlüsse des 8080 sind über die Treiber-Bausteine 8216 geschaltet. Die beiden Bausteine sind so geschaltet, daß wir einen direkten Ein/Ausgangs-Betrieb realisieren können. Die Informationen des Daten-BUS liegen an der Speicher- und an den Eingabe- und Ausgabe-Einheiten des Mikroprozessor-Systems an. Mit den einzelnen Funktionen können wir Einschreiben, Auslesen und ein Programm ablaufen lassen.

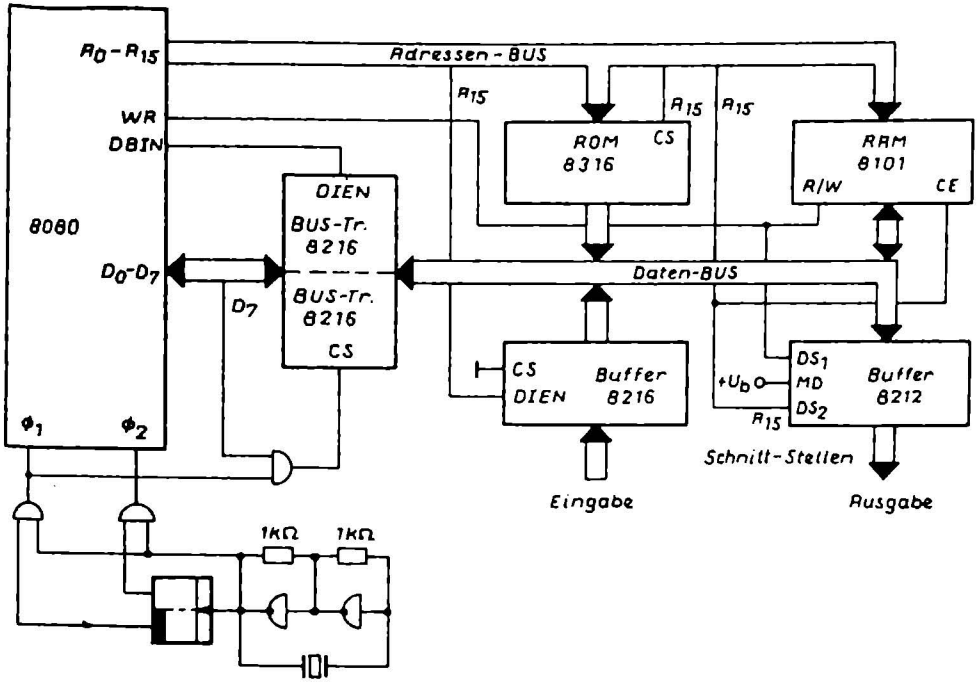


Bild 2.4.11 Aufbau einer einfachen Mikroprozessor-Schaltung

Ein Standard-System für ein kleines Mikroprozessor-System ist in Bild 2.4.12 gezeigt.

Mit dem MEM R-Ausgang der System-Steuerung werden die ROM- und die RAM-Bausteine angesteuert. Ein L-Signal auf der MEM R-Leitung und die Daten des ROMs werden ausgelesen. Das RAM dagegen kann nur dann mitansprechen, wenn der MEM W-Ausgang der System-Steuerung ein L-Signal hat.

Mit seinen drei Steuer-Ausgängen WR, DBIN und HLDA steuert der Mikroprozessor 8080 den Baustein 8228 an. Der WR-Ausgang dient zum Schreiben, der DBIN-Ausgang für die Daten-BUS-Steuerung und der HLDA-Ausgang für die Halt Steuerung. Zusammen mit dem STSTB-Eingang des 8228 ergeben sich die Zustandskombinationen für die Steuerung der Speicher- und Ein/Ausgabe-Einheiten.

Die Daten der Ein/Ausgabe/Einheit können auf den Daten-BUS gelangen, wenn die I/O R-Leitung ein L-Signal hat. Die Daten des Daten-BUS werden auf den Ausgang des Systems geschaltet, wenn die I/O W-Leitung ein L-Signal hat.

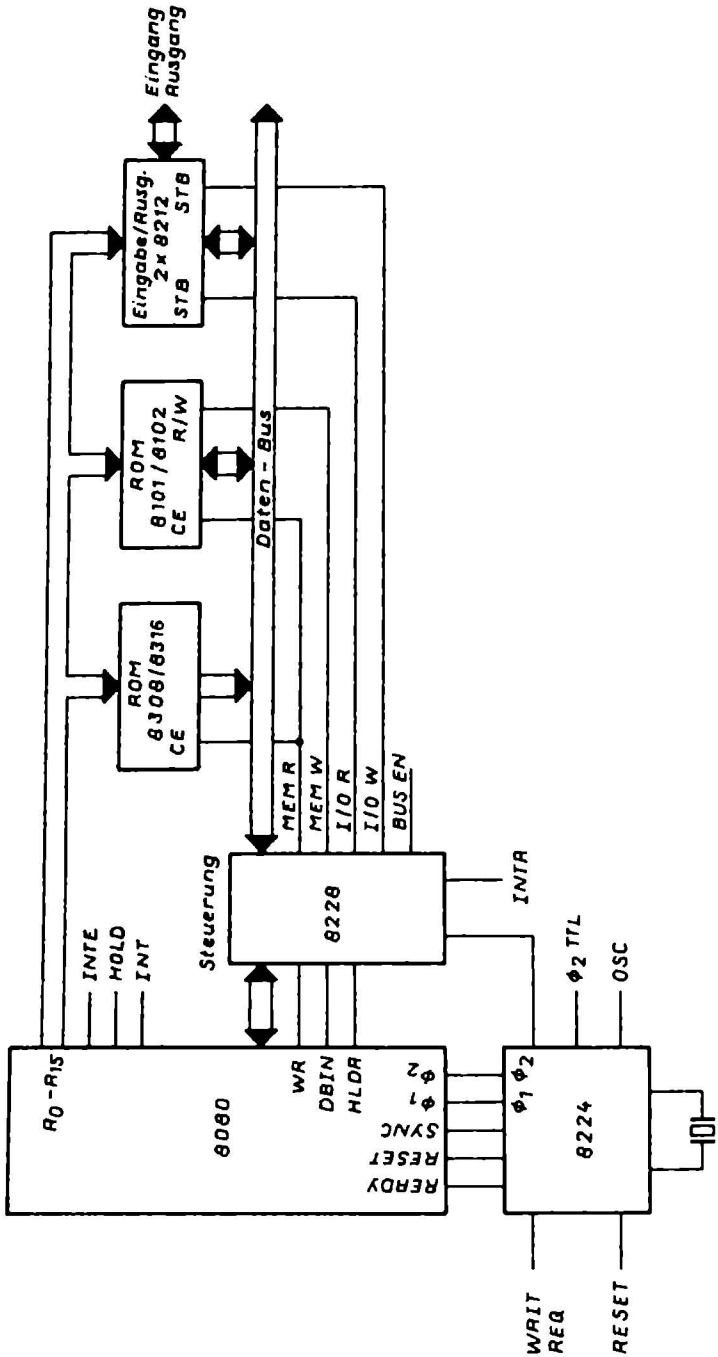


Bild 2.4.12 Standard-System für ein kleines Mikroprozessor-System

Befehlsliste

Mnemonic	Binär-Code	Beeinfl. Zustands-Bits	Englische Befehlsschreibung
Transferbefehle			
Register Register			
MOV r_1, r_2	L H D D D S S S	– – – –	Move register to register
XCHG	H H H L H L H H	– – – –	Exchange D&E, H&L
XTHL	H H H L L L H H	– – – –	Exchange top of stack H&L
SPHL	H H H H H L L H	– – – –	H&L to stak pointer
Speicher, Peripherie Register			
MOV r_1, M	L H D D D H H L	– – – –	Move memory to register
LDA adr	L L H H H L H L	– – – –	Load accumulator direct
LDAX rp1	L L R P H L H L	– – – –	Load accumulator indirect
LHLD adr	L L H L H L H L	– – – –	Load H&L direct
POP rp1	H H R P L L L H	– – – –	Pop register pair rp off stack
IN nr	H H L H H L H H	– – – –	Input
Konstante Registerpaar			
LXI rp1, adr	L L R P L L L H	– – – –	Load register pair immediate
Register Speicher, Peripherie			
MOV M, r1	L H H H L S S S	– – – –	Move register to memory
STA adr	L L H H L L H L	– – – –	Store accumulator direct
STAX rp1	L L R P L L H L	– – – –	Store accumulator indirect
SHLD adr	L L H L L L H L	– – – –	Store H&L direct
PUSH rp1	H H R P L H L H	– – – –	Push register pair rp on stack
OUT	H H L H L L H H	– – – –	Output
Konstante Register-Speicher			
MVI M, konst	L L H H L H H L	– – – –	Move to memory immediate
MVI r1, konst	L L D D D H H L	– – – –	Move immediate Register

Funktion des Befehls
r_1, r_2=A, B, C, D, H oder L Lade Register r_1 mit dem Inhalt v. Register r_2
Vertausche Inhalte der Registerpaare (D, E) und (H,L)
Vertausche Inhalt des Registerpaares (H,L) und den Inhalt des Wortes, das durch den Stackpointer adressiert ist
Lade Stackpointer mit dem Inhalt des Registerpaares H, L
r_1=A, B, C, D, E, H oder L Lade Register r_1 mit dem Inhalt des Speicherbytes, das durch den Inhalt des Registerpaares (H, L) adressiert ist
Akkumulator laden mit Inhalt der Adresse adr
rp=B, C oder D, E Akkumulator laden mit Inhalt des Speicherplatzes, der durch den Inhalt des Registerpaares rp1 adressiert ist
Lade Registerpaar (H,L) mit dem Inhalt der Adressen adr und (adr + 1)
rp1=B, C; D,E oder H,L Registerpaar rp1 wird mit dem Wort geladen, das durch den Stapelzeiger adressiert ist
Akkumulator wird mit dem Inhalt des Eingabekanals Nummer nr (nr $\geq$255) gelad.
rp1=B, C; D,E; H,L Lade Registerpaar rp1 mit Wert adr
r_1=A, B, C, D, E, H oder L Inhalt von Register r_1 auf den Speicherplatz abspeichern, der durch den Inhalt des Registerpaares (H, L) adressiert ist
Akkumulator Inhalt unter Adresse adr abspeichern
rp1=B, C oder D, E Akkumulator in dem Byte abspeichern, das durch den Inhalt des Registerpaares rp1 adressiert ist
Registerpaar unter Adresse adr u. (adr + 1) abspeichern
rp1=PSW, B, C; D, E oder H, L Inhalt des Registerpaares rp1 wird in das Wort übertragen, das durch den Stapelzeiger adressiert ist
Akkumulator-Inhalt wird auf Ausgabekanal (Nummer nr$\geq$255) ausgegeben
Bringe den Wert des Konstanten (konst.$\geq$255) auf den Speicherplatz, der durch den Inhalt des Registerpaares (h, L) adressiert ist
r1=A, B, C, D, E, H, L Lade Register r1 mit Konstante (konst.$\geq$255)

Mnemonic	Binär-Code	Beeinfl. Zustands-Bits	Englische Befehlsschreibung

SPRUNGBEFEHLE

Unbedingte Sprünge

PCHL	H H H L H L L H	– – – –	H&L to programm counter
JMP adr	H H L L L L H H	– – – –	Jump unconditional

Bedingte Sprünge

JC adr	H H L H H L H L	– – – –	Jump on no carry
JNC adr	H H L H L L H L	– – – –	Jump on no carry
JZ adr	H H L L H L H L	– – – –	Junp on zero
JNZ adr	H H L L L L H L	– – – –	Jump on no zero
JM adr	H H H H H L H L	– – – –	Jump on minus
JP adr	H H H H L L H L	– – – –	Jump on positiv
JPE	H H H L H L H L	– – – –	Jump on parity even
JPO	H H H L L L H L	– – – –	Jump on parity odd
RST konst.	H H N N N H H H	– – – –	Restart

UNTERPROGRAMMBEHANDLUNG

Programmaufrufe

CALL adr	H H L L H H L H	– – – –	Call unconditional
Bei allen Aufrufbefehlen wird die Rückkehradresse im Stapelspeicher abgelegt			
CC adr	H H L H H H L L	– – – –	Call on carry
CNC adr	H H L H L H L L	– – – –	Call on no carry
CZ adr	H H L L H H L L	– – – –	Call on zero
CNZ adr	H H L L L H L L	– – – –	Call on no zero
CM adr	H H H H H H L L	– – – –	Call on minus
CP adr	H H H H L H L L	– – – –	Call on positiv
CPE adr	H H H L H H L L	– – – –	Call on parity even
CPO adr	H H H L L H L L	– – – –	Call on parity odd

Funktion des Befehls
Programm wird auf der Adresse fortgesetzt, die im Registerpaar (H.L) steht
Programm wird bei adr fortgesetzt
Bei Carry-Bit=H wird das Programm bei der Adresse adr fortgesetzt
Bei Carry-Bit=L wird das Programm bei der Adresse adr fortgesetzt
Bei Zero-Bit=H wird das Programm bei der Adresse adr fortgesetzt
Bei Zero-Bit=L wird das Programm bei der Adresse adr fortgesetzt
Bei Sign-Bit=H wird das Programm bei der Adresse adr fortgesetzt
Bei Sign-Bit=L wird das Programm bei der Adresse adr fortgesetzt
Bei Paritäts-Bit=H wird das Programm bei der Adresse adr fortgesetzt
Bei Paritäts-Bit=L wird das Programm bei der Adresse adr fortgesetzt
Ø≥konst ≥ Programm wird auf der Adresse 8 x konst. fortgesetzt, die Rückkehr-adresse wird im Stapelspeicher abgelegt
Progr.wird bei der Adresse adr fortgesetzt, Rückkehradresse wird im Stapel abgelegt
Bei Carry-Bit=H wird das Programm bei der Adresse adr fortgesetzt
Bei Carry-Bit=L wird das Programm bei der Adresse adr fortgesetzt
Bei Zero-Bit=H wird das Programm bei der Adresse adr fortgesetzt
Bei Zero-Bit=L wird das Programm bei der Adresse adr fortgesetzt
Bei Sign-Bit=H wird das Programm bei der Adresse adr fortgesetzt
Bei Sign-Bit=L wird das Programm bei der Adresse adr fortgesetzt
Bei Paritäts-Bit=H wird das Programm bei der Adresse adr fortgesetzt
Bei Paritäts-Bit=L wird das Programm bei der Adresse adr fortgesetzt

Mnemonic	Binär-Code	Beeinfl. Zustands-Bits	Englische Befehlsschreibung
LOGISCHE OPERATIONEN			
CMA	L L H L H H H H	- - - -	Complement accumulator
ANA r1	H L H L L S S S	Z,S,P CY,AC	And register with accumulator
ANA M	H L H L L H H L	Z,S,P,CY,AC	And memory with accumulator
ANI konst.	H H H L L H H L	Z,S,P,CY,AC	And immediate with accumulator
ORA r1	H L H H L S S S	Z,S,P CY,AC	Or register with accumulator
ORA M	H L H H L H H L	Z,S,P, CY,AC	Or memory with accumulator
ORI konst	H H H H L H H L	Z,S,P,CY,AC	Or immediate with accumulator
XRA r1	H L H L H S S S	Z,S,P CY,AC	Exclusive Or register with accumulator
XRA M	H L H L H H H L	Z,S,P, CY,AC	Exclusive Or memory with accumulator
XRI konst	H H H L H H H L	Z,S,P, CY,AC	Exclusive Or immediate with accumulator
CMP r1	H L H H H S S S	Z,S,P, CY,AC	Compare register with accumulator
CMP M	H L H H H H H L	Z,S,P, CY,AC	Compare memory with accumulator
CPI konst.	H H H H H H H L	Z,S,P CY,AC	Compare immediate with accumulator
REGISTERANWEISUNGEN			
Akkumulator rotieren			
RLC	L L L L L H H H	- - -CY-	Rotate accumulator left
RRC	L L L L H H H H	- - -CY-	Rotate accumulator right
RAL	L L L H L H H H	- - -CY-	Rotate accumulator left through carry
RAR	L L L H H H H H	- - -CY-	Rotate accumulator right through carry
Übertragsbit-Anweisungen			
CMC	L L H H H H H H	- - -CY-	Complement carry
STC	L L H H L H H H	- - -CY-	Set carry

Funktion des Befehls
Akkumulator-Inhalt wird negiert
r1=A,B,C,D,E,H oder L Akkumulator-Inhalt und der Inhalt des Registers r1 werden UND-verknüpft
Der Inhalt des durch Registerpaar (H,L) adressierten Bytes wird m. UND verknüpft
Akkumulator-Inhalt wird mit der konst ≥255 UND-verknüpft
r1=A,B,C,D,E,H oder L Akkumulator-Inhalt wird mit dem Inhalt des Registers r1 ODER-verknüpft
Inhalt des über Registerpaar (H,L) adressierten Bytes wird mit Akkumulator-Inhalt ODER-verknüpft
Akkumulator-Inhalt wird mit konst.≥255 ODER-verknüpft
r1=A,B,C,D,E,H oder L Akkumulator-Inhalt wird mit dem Inhalt des Registers r1 Exclusiv-ODER-verknüpft
Das über Registerpaar (H,L) adressierte Byte wird durch Exclusive-ODER mit dem Akkumulator verknüpft
Der Akkumulator wird m. dem Wert konst.≥255 durch Excl.-ODER verknüpft
r1=A,B,C,D,E,H oder L Akkumulator-Inhalt wird mit dem Inhalt des Registers r1 verglichen
Akkumulator-Inhalt wird mit dem Inhalt des durch Registerpaar (H,L) adressierten Bytes verglichen
Akkumulator-Inhalt wird mit konst.≥255 verglichen
Akkumulator-Inhalt wird zyklisch um 1 Bit nach links verschoben. Bit 7 wird in das Carry-Bit geschrieben
Akkumulator-Inhalt wird zyklisch um 1 Bit nach rechts verschoben. Bit 0 wird in das Carry-Bit geschrieben
Akkumulator-Inhalt und das Carry-Bit werden nach links verschoben
Akkumulator-Inhalt und das Carry-Bit werden nach rechts verschoben
Carry-Bit wird negiert
Carry-Bit wird gesetzt

Mnemonic	Binär-Code	Beeinfl. Zustands-Bits	Englische Befehlsschreibung
ARITHMETISCHE OPERATIONEN			
INR r1	L L D D D H L L	Z,S,P,AC	Increment register
INR M	L L H H L H L L	Z,S,P,AC	Increment memory
DCR r1	L L D D D H L H	Z,S,P,AC	Decrement register
DCR M	L L H H L H L H	Z,S,P,AC	Decrement memory
INX r1	L L R P L L H H	- - - -	Increment register pair
DCX rp1	L L R P H L H H	- - - -	Decrement register pair
ADD r1	H L L L L S S S	Z,S,P, CY, AC	Add register to accumulator
ADD M	H L L L L H H L	Z,S,P CY,AC	Add memory to accumulator
ADC r1	H L L L H S S S	Z,S,P CY,AC	Add register to accumulator with carry
ADC M	H L L L H H H L	Z,S,P CY,AC	Add memory to accumulator with carry
DAD rp1	L L R P H L L H	- -CY-	add register pair to H and L
SUB r1	H L L H L S S S	Z,S,P CY,AC	subtract register from accumulator
SUB M	H L L H L H H L	Z,S,P CY,AC	Subtract memory from accumulator
SBB r1	H L L H H S S S	Z,S,P CY,AC	Subtract register from accumulator with borrow
SBB M	H L L H H H H L	Z,S,P, CY,AC	Subtract memory from accumulator with borrow
ADI konst.	H H L L L H H L	Z,S,P,CY,AC	Add immediate to accumulator
ACI konst.	H H L L H H H L	Z,S,P, CY,AC	Add immediate to accumulator with carry
SUI konst.	H H L H L H H L	Z,S,P CY,AC	Subtract immediate from accumulator
SBI konst.	H H L H H H H L	Z,S,P CY,AC	Subtract immediate from accumulator with borrow
DAA	L L H L L H H H	Z,S,P,CY,AC	Decimal adjust accumulator

Funktion des Befehls
r_1=A,B,C,D,E,L oder H Zum Inhalt des Registers r1 wird 1 addiert
Zum Inhalt des durch das Registerpaar (H,L) adressierten Bytes wird 1 addiert
r_1=A,B,C,D,E,H oder L Vom Inhalt des Registers r1 wird 1 subtrahiert
Vom Inhalt des durch das Registerpaar (H,L) adressierten Bytes wird 1 subtrahiert
rp1=B,C; D,E oder H,L Der Inhalt des Registerpaares rp1 wird um 1 erhöht
rp1=B,C; D,E oder H,L Der Inhalt des Registerpaares rp1 wird um 1 erniedrigt
r1=A,B,C,D,E,H oder L Inhalt von Register r1 wird zum Inhalt des Akkumulators addiert
Der Inhalt des Speicherbytes, das durch den Inhalt des Registerpaares (H,L) adressiert ist, wird zum Inhalt des Akkumulators addiert
r1=A,B,C,D,E,H oder L Inhalt des Registers r1 und Inhalt des Carry-Bits werden zum Inhalt des Akkumulators addiert
Inhalt des Speicherbytes, das durch den Inhalt des Registerpaares (H,L) adressiert ist, und der Inhalt des Carry-Bits werden zum Inhalt des Akkumulators addiert
rp1=B,C; D, E oder H, L Inhalt des Registerpaares rp1 und der Inhalt des Registerpaares (H,L) werden addiert. Ergebnis in (H,L)
r1=A,B,C,D,E,H oder L Inhalt des Registers r1 wird vom Akkumulator-Inhalt subtrahiert
Inhalt des Speicherbytes, das durch das Registerpaar (H,L) adressiert ist, wird vom Akkumulator subtrahiert
r1=A,B,C,D,E,H oder L Inhalt von Register r1 und Inhalt des Carry-Bits werden vom Akkumulator-Inhalt subtrahiert
Inhalt des Speicherbytes, das durch das Registerpaar (H,L) adressiert ist, und Inhalt des Carry-Bits werden vom Akkumulator-Inhalt subtrahiert
Konstante (konst.≥255) wird zum Inhalt des Akkumulators addiert
Zum Akkumulator-Inhalt werden konst.≥255 und Carry-Bit addiert
konst.≥255 wird vom Akkumulator-Inhalt subtrahiert
konst.≥255 und das Carry-Bit werden vom Akkumulator-Inhalt subtrahiert
Akkumulator-Inhalt wird in eine 2stellige Zahl umgewandelt

Mnemonic	Binär-Code	Beeinfl. Zustands-Bits	Englische Befehlsschreibung
Rücksprungbefehle			
RET	H H L L H L L H	- - - -	Return
RC	H H L H H L L L	- - - -	Return on carry
RNC	H H L H L L L L	- - - -	Return on no carry
RZ	H H L L H L L L	- - - -	Return on zero
RNZ	H H L L L L L L	- - - -	Return on no zero
RM	H H H H H L L L	- - - -	Return on minus
RP	H H H H L L L L	- - - -	Return on positiv
RPE	H H H L H L L L	- - - -	Return on parity even
RPO	H H H L L L L L	- - - -	Return on parity odd
PROGRAMMUNTERBRECHUNG			
EI	H H H H H L H H	- - - -	Enable interrupt
DI	H H H H L L H H	- - - -	Disable interrupts
SONSTIGE BEFEHLE			
HLT	L H H H L H H L	- - - -	Halt
NOP	L L L L L L L L	- - - -	No operation

Funktion des Befehls
Programm wird auf der Adresse fortgesetzt, die in dem Wort steht, das über den Stapelzeiger adressiert ist.
Bei Carry-Bit=H wird das Programm auf der Adresse fortgesetzt, die in dem über den Stapelzeiger adressierten Wort steht.
Bei Carry-Bit=L wird das Programm auf der Adresse fortgesetzt, die in dem über dem Stapelzeiger adressierten Wort steht.
Bei Zero-Bit=H wird das Programm auf der Adresse fortgesetzt, die in dem über dem Stapelzeiger adressierten Wort steht.
Bei Zero-Bit=L wird d as Programm auf der Adresse fortgesetzt, die in dem über dem Stapelzeiger adressierten Wort steht.
Bei Sign-Bit=H wird das Programm auf der Adresse fortgesetzt, die in dem über dem Stapelzeiger adressierten Wort steht.
Bei Sign-Bit=L wird das Programm auf der Adresse fortgesetzt, die in dem über dem Stapelzeiger adressierten Wort steht.
Bei Paritäts-Bit=H wird das Programm auf der Adresse fortgesetzt, die in dem über dem Stapelzeiger adressierten Wort steht.
Bei Paritäts-Bit=L wird das Programm auf der Adresse fortgesetzt, die in dem über dem Stapelzeiger adressierten Wort steht.
INTE-Flip-Flop wird gesetzt; d. Mikroprozessor kann Unterbrechungen annehmen
INTE-Flip-Flop wird rückgesetzt; der Mikroprozessor ignoriert Unterbrechungsanforderungen
Programm hält an, bis eine Unterbrechungsanforderung eintrifft
Leerbefehl

Best.Nr.

1

TBB - Handbuch Band 1 , W. Hofacker

Transistor Berechnungs- und Bauanleitungshandbuch Band 1

Das Handbuch für jeden Elektroniker. Rechenbeispiele, Berechnungsgrundlagen, Bauanleitungen, Nomogramme, Tabellen und Vergleichslisten aus den wichtigsten Bereichen der Elektronik. Ein Buch zum Einarbeiten in die Elektronik. Ein Buch zum Nachschlagen. Grundlagen Digitaltechnik, Netzgeräte und Transformatorenberechnung, Berechnung von Multivibratoren, Schmitt Trigger u. v. a. Über 130 Seiten. **DM 19,80**

2

TBB - Handbuch Band 2 , W. Hofacker

Transistor Berechnungs- und Bauanleitungshandbuch Band 2

Dieses Buch ist die Fortsetzung des erfolgreichen Handbuches TBB-Handbuch Band 1. Ein Buch, das sich in der Hand des Praktikers bestens bewährt hat. Weitere neueste Schaltbeispiele und Berechnungsgrundlagen. Experimentier- und Versuchsbeschreibungen. Integrierte Spannungsregler, Wärmeableitung, Operationsverstärker Einführung, RC-Zeitglieder, Transistortester u. v. a. **DM 19,80**

3

Elektronik im Auto , H. Gebauer

Ein Buch für jeden technisch interessierten Autofahrer. Es zeigt Ihnen die vielen Möglichkeiten zur Verbesserung von Sicherheit, Leistung und Fahrkomfort in Ihrem Auto. Thyristorzündung, Drehzahlmesser, Beschleunigungsmesser, Geschwindigkeitswarner, Batterieladegerät u. v. a. Tips, genaue Beschreibungen, Bauanleitungen. **DM 9,80**

4

IC - Handbuch , C. Lorenz

Handbuch für digitale und lineare integrierte Schaltungen.

Sensationelle Neuheit. Ein Handbuch für digitale und lineare Schaltkreise. Daten- und Auswahllisten, Vergleichslisten, Gehäuseformen, Grundlagen, Einführungsbeschreibungen, viele Schaltbeispiele, Bauanleitungen, Printvorlagen, u. v. a.
Alles über TTL-Technik, C MOS, MOS-Schaltungen, Uhren ICs, lineare Schaltungen, ein Chip-Rechner, Integrierte NF-Verstärker u. v. a. **DM 19,80**

5

IC - Datenbuch , D. Steinbach

Daten- und Auswahllisten der gebräuchlichsten integrierten Schaltkreise. Digital und analog. Gerade bei ICs ist es wichtig die Anschlußfolgen genau zu kennen. Auf über 55 Seiten finden Sie: Die wichtigsten TTL-Schaltkreise, C-Mos Serie, lineare Schaltungen wie Operationsverstärker, Komparatoren, NF-Verstärker, Spannungsregler, Triggerschaltungen, Impulsgeber u. v. a. Weiterhin finden Sie eine C Mos-Vergleichsliste sowie Kurzdaten und logisches Verhalten dieser C MOS Elemente. Das IC-Datenbuch wird auch Ihnen ein unentbehrlicher Begleiter bei allen Arbeiten mit integrierten Schaltungen sein. **DM 9,80**

6

IC - Schaltungen , D. Steinbach

Hier finden Sie eine gelungene Zusammenstellung der wichtigsten Anwendungsbeispiele aus dem Bereich der integrierten Schaltungen. TTL – C MOS – Linear. Alle Schaltungen sind übersichtlich und klar dargestellt und mit einer kurzen, jedoch sehr genauen Beschreibung versehen. Viele Schaltungen sind Grundschaltungen, die man beim Umgang mit integrierten Schaltungen immer wieder benötigt. Tastenentprellung, Zähler, Impulsgeber, Codierer, Dekodierer, Datenübertragung, Serien-Parallel-Wandler, Digitalvoltmeter u. v. a. **DM 9,80**

7

Elektronik Schaltungen,
4. völlig neu überarbeitete Auflage, W. Hofacker

Die ideale Schaltungssammlung zum Basteln und Experimentieren. Schaltungen mit Operationsverstärkern, Spannungsreglern, TTL, C-MOS Schaltkreisen. MOS Uhr mit Wecker, elektronischer Würfel, Musik Synthisizer, Timer 555 Anwendungen, Experimentieranleitungen und viele andere hochinteressante Schaltbeispiele tlw. mit Printvorlage. 64 Seiten Inhalt. **DM 5,–**

8

IC - Bauanleitungs -Handbuch -IC -KIT , C. Lorenz

Ein Bauanleitungsbuch mit vielen hochinteressanten Bauanleitungen aus dem Bereich der LSI Schaltungstechnik. Schaltbeisplele mit Printvorlagen zum Selbstherstellen der Leiterplatten mit genauesten Beschreibungen. Hochaktuell und brandneu: Funktionsgenerator XR 2206, MOS-Uhr mit Wecker, Schlummerautomatik und programmierbarem Wecktongenerator, Sensortastenwahl, IC-Netzteil, Funktionsgenerator 8038 neuartige Transistorzündung, 35 W NF-Verstärker, Experimentieranleitung und Grundkurs über Flip Flops, Experimente mit Digitalschaltungen u. v. a. Zu allen Schaltungen finden Sie Platinvorlagen oder Sie können die Experimentierschaltungen auf der Experimentierplatine WH-1 g durchführen. Über 125 Seiten. **DM 19,80**

41

Experimentierplatine mit Sockel, Stecker und Füßen Typ WH-1g für 40,28,24,16 und 14 polige DIL -Gehäuse **DM 79,–**

9

Feldeffekttransistoren, C. Lorenz

Der Feldeffekttransistor (FET) gehört heute zu den interessantesten Bauteilen überhaupt. Wie man damit experimentiert, wie man seine Funktion versteht und wie man damit brauchbare und hochinteressante Schaltungen aufbauen kann, zeigt Ihnen dieses Buch. Grundlagen, Kennlinienfelder, Tabellen, Berechnungsgrundlagen, Rechenbeispiele, Anschlußbilder und eine Vergleichsliste für Feldeffekttransistoren bilden den Kern dieser umfangreichen Darstellung. Alles in allem finden Sie hier eine praxisnahe und komplette Arbeitsunterlage, mit der Sie im Beruf und auch im Hobby erfolgreich arbeiten können. Über 45 Seiten. **DM 5,–**

10

Elektronik und Radio, C. Lorenz
4. Auflage. Völlig neu bearbeitet und stark erweitert.

Eine Einführung in die Radiotechnik, wie man sie nicht alle Tage findet. Eine sehr geschickt gemachte Einführung mit vielen Schaltungen, Bauanleitungen und genauesten Funktionsbeschreibungen. Vom einfachen Diodenempfänger (Detektor) bis zu interessanten Sender- und Empfängerschaltungen. (Minispione) Viele hundert Bilder zeigen Ihnen genau, wie Sie beim Experimentieren vorgehen müssen. IC-Radio, IC-Sender, Antennen, Berechnungsgrundlagen, Tabellen u. v. a. Über 150 Seiten **DM 19,80**

11

IC -Niederfrequenzverstärker , C. Lorenz

Grundlagen der integrierten NF-Verstärker, Berechnung von kompletten IC-NF-Verstärkerstufen. Anwendungsbeispiele mit den interessantesten und gebräuchlichsten Standard IC-NF-Verstärkern wie TBA 800, TBA 830, usw. Printvorlagen, Auswahltabellen, Experimentieranleitungen und Anschlußbilder machen dieses Buch zu einem unentbehrlichen Begleiter für alle, die sich mit NF-Verstärkern beschäftigen wollen. Über 65 Seiten. **DM 9,80**

12

BIS BUCH, Beispiele integrierter Schaltungen, H. Bernstein

Auf über 130 Seiten Anwendungsbeispiele mit integrierten Schaltkreisen. Zeitgeber 555, Funktionsgenerator ICL 8038, Opto Elektronik, Operationsverstärker, Analogschalter, Digital-Analog-Wandler, Analoge Rechenbausteine, Schreib-Lese-Speicher (RAM), Festwertspeicher (ROM), Speicherschaltungen, Uhrenbausteine u. v. a. **DM 19,80**

13

HEH, Hobby Elektronik Handbuch , C. Lorenz

Das Schaltungsbuch für jeden Hobbyelektroniker. Schaltbeispiele und Bauanleitungen aus dem gesamten Hobbybereich. Lichtorgeln, Eiswarngerät fürs Auto, Alarmanlagen, Metallsuchgerät, PLL-Schaltungen, Logik-Tester, Funktionsgeneratoren u. v. a. Über 55 Seiten. **DM 9,80**

14

IC - Vergleichsliste , C. Lorenz

Vergleichsliste für digitale und lineare integrierte Schaltkreise.

Standard TTL, Low Power Schottky TTL, C MOS, Triacs Thyristoren, Optoelektronik, Operationsverstärker, Spannungskomparatoren, Spannungsregler, NF-Verstärker u. v. a. Funktionsvergleichsliste CMOS zu TTL. Vergleichstabelle für Transistoren und Dioden sowie Darlingtontransistoren. Eine Vergleichsliste, die man immer wieder braucht. **DM 9,80**

15

Opto -Handbuch, Handbuch für Optoelektronik , C. Lorenz

Das Handbuch für die gesamte Optoelektronik. Eine Einführung und ein ideales Nachschlagwerk. Grundlagen, Definitionen aller Kenngrößen, Opto-Lexikon, Berechnungsgrundlagen, Rechenbeispiele, Schaltbeispiele: Lichtsender, Lichtempfänger, Anzeigen, Infrarot Detektoren, Lichtmeßgerät, Optokoppler, Pegelschalter, Opto-Vergleichsliste. Anschlußbilder wichtiger 7-Segment-Anzeigen u. v. a. Über 106 Seiten. **DM 19,80**

16

C MOS Einführung, Entwurf, Schaltbeispiele, Teil 1 , H. Bernstein

Vom C MOS Gatterbaustein über Schieberegister und Zähler bis hin zum C MOS Schreib-Lesespeicher. Insgesamt werden neunzehn interessante und bekannte C MOS Schaltkreise beschrieben. Zu jedem Bauelement sind genaue Daten, Schaltbild und Anwendungsbeispiele angegeben. Im großen Applikationsteil finden Sie: C MOS-Kippstufen, Addierwerke und Rechenschaltungen, Digital Analog Wandler, Schieberegister für analoge Spannungen, Multiplexsysteme für analoge Signale u. v. a. Eine komplette Einführung und gut geeignet für das Selbststudium der C MOS Technik. Über 140 Seiten. **DM 19,80**

17

C MOS Entwurf und Schaltbeispiele, Teil 2 , H. Bernstein

Fortsetzung von Teil 1. Anwendungsbeispiele mit genauen Schaltungsbeschreibungen und Bauelementeunterlagen. Daten, Anschlußbelegungen weiterer wichtiger hochintegrierter C MOS Elemente. Ein komplettes Arbeits- und Experimentierbuch. C-MOS Uhrenschaltungen, Schieberegisterschaltungen, Parallel-Serien Umsetzung, statische und dynamische Speicherschaltungen, Zählschaltungen, Digital Analog Wandler, Analog Digital Wandler, Digital Voltmeter, I/O Registerschaltungen, Codier und Dekodierschaltungen. RAM und ROM Anwendungen. Über 140 Seiten. **DM 19,80**

18

C MOS Entwurf und Schaltbeispiele, Teil 3 , H. Bernstein

Fortsetzung von Teil 2. Eine sehr umfangreiche Applikationssammlung mit hochintegrierten C MOS Elementen. Rechnerschaltungen, Speicher- und Steuerschaltungen, Multiplex- und Datenbussysteme, Uhrenschaltungen, PLL-Schaltungen, Liquid Cristal Anzeigen und deren Treiberschaltungen, Optoelektronik in Verbindung mit C MOS. Grundlagen, Aufbau und Wirkungsweise der Prozeßrechentechnik, Arithmetische Logische Einheiten (ALU) und andere wichtige Funktionen aus der Prozeßrechentechnik. RAMs, ROMs und FIFO-Speicherschaltungen. Über 140 Seiten. **DM 19,80**

19

IC Experimentier Handbuch - IC -EX, C. Lorenz C. Lorenz

Eine sehr umfangreiche Schaltungssammlung und Bauanleitunssammlung mit neuesten integrierten Bausteinen. Neue, jedoch beim Fachhandel erhältliche Standard ICs. Rechnerschaltungen, Mikroprozessoren, I/O Schaltungen, druckende und anzeigende Rechner, Stoppuhren, Zählerschaltungen, Digitalvoltmeter, professioneller Synthisizer, Hilfsschaltungen für den Elektronik Experimenter, Analog Digital Wandler, Frequenzzähler u. v. a. hochinteressante Bauanleitungen. Viele Schaltungen können auf der IC KIT Experimentierplatine WH–1g aufgebaut werden. Über 120 Seiten. **DM 19,80**

20

Operationsverstärker, Grundlagen und Schaltbeispiele , C. Lorenz

Dieses Buch umfaßt das gesamte Gebiet der linearen Schaltungstechnik und stellt ein in dieser Preislage bisher noch nie dagewesen Nachschlagwerk und Einführungshandbuch dar. Bestens geeignet für das Selbststudium. Nach einer pädagogisch geschickt gemachten Einführung folgen theoretische Arbeitsunterlagen und die zugehörigen Schaltbeispiele mit Daten und Gehäuseanschlüssen. Dieses wertvolle Buch dürfte seinen Platz auch bei Ihren Arbeitsunterlagen finden, und wird dann immer von Nutzen sein, wenn es um die Lösung von nicht routinemäßigen Aufgaben geht. **DM 19,80**

21

Digitaltechnik Grundkurs (TTL -C MOS - MOS und Software) , C. Lorenz

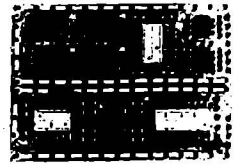
Ein Einführungskurs in die Digitaltechnik für Anfänger und Fortgeschrittene. Ein Fachbuch für den programmierten Selbstunterricht. Der ideale Kurzlehrgang für das Selbststudium. Der Kurs vermittelt Ihnen alle wichtigen Grundkenntnisse vom TTL-Gatter bis zum Mikroprozessor und Lösung von Schaltungsaufgaben durch Software. Viele Versuchsaufbauten und Experimente aus diesem Kurs können auf der IC-KIT Platine WH–1g durchgeführt werden. Grundlagen, Gatter, Zähler, programmierbare Zähler, IC-Tester, Schieberegister, Speicher, Mikroprozessoren u. v. a. Über 130 Seiten. **DM 19,80**

41

Experimentierplatine WH-1g dazu, Best.Nr. 41 **DM 79,–**

Experimentierplatine WH - 1 g
Abmessungen: 210 mm x 150 mm

Experimentierplatine WH - 1 g
fertig aufgebaut und mit Sockeln bestückt.



22

Mikroprozessoren, Eigenschaften und Aufbau Teil 1 , H. Bernstein

Grundlagen, Eigenschaften und Aufbau von Mikroprozessoren. Organisation von Recheneinheiten und Mikroprogrammen. Programmierung und Klassifizierung von Mikroprozessoren. Ablaufdiagramm, Flußdiagramm. Ein Chip-Technik und Multi Chip-Technik, Transfer- und Sprungfunktionen. Speichertechnik: RAMs, ROMs, FIFO, FILO. Programmierbare logische Arrays (PLA)
Anwendungsbeispiele und Anwendungsbereiche. Über 120 Seiten. **DM 19,80**

23

Elektronik Grundkurs (Kurzlehrgang Elektronik) , C. Lorenz

Eine leichtverständliche und pädagogisch geschickt gemachte Einführung in die Technik der elektronischen Schaltungen. Ein Kurzlehrgang und Schnellkurs zugleich. Aber auch ein recht brauchbares Nachschlagewerk für den fortgeschrittenen Elektroniker. Mit wenig Mühe können Sie sich hier die Grundkenntnisse der elektronischen Schaltungspraxis aneignen. Das Buch schafft die Voraussetzungen für ein erfolgreiches und sicheres Arbeiten mit interessanten Schaltkreisen modernster Technologien. Unentbehrlich für das Experimentieren mit den heutigen modernen hochintegrierten Schaltkreisen. Über 150 Seiten.
DM 9,80

24

Mikrocomputer Technik, Hans Peter Blomeyer-Bartenstein

In diesem Buche finden Sie eine umfassende, einführende und weiterführende Hilfe zum Einstieg in die Mikrocomputertechnik mit vielen Schalt- und Programmierbeispielen. Als praktische Betrachtungsgrundlage dient das supermoderne Microcomputerkonzept Z80A von ZILOG. Das Buch geht auf alle wichtigen Zusammenhänge ein und erklärt diese dem Leser so ausführlich, daß kaum noch Fragen offen bleiben. Über 240 Seiten **DM 29,80**

25

Hobby Computer Handbuch, C. Lorenz

Eine leicht verständliche Einführung in die Microcomputertechnik.

Diese sehr umfangreiche Einführung in die Microcomputertechnik dürfte zu diesem Preise einmalig sein. Auf über 450 Seiten finden Sie: Grundlagen der Computer- und Microcomputer-Technik, Was ist ein Microcomputer?, Microcomputer KITs, Einplatinencomputer, KIM, MIKIT, Z80 KIT, 6800 KIT, NEC 8080 KIT u. v. a. Genaue Beschreibungen der wichtigsten Mikroprozessortypen. Zusammenstellung und Beschreibung der modernen Personal Computer. (IMSAI, CROMEMCO, CAT, OSI, POLY 88 u. v. a.) Kompaktcomputer wie PET und TR-8. Interfactechniken, Ein/Ausgabegeräte, ROMs, RAMs, Programmiergeräte für PROMs. Löschgeräte für EPROMs u. v. a. mehr. Das ideale allumfassende Buch für den Microcomputertechniker. Für Industrieanwendung ebenso geeignet, wie für den Hobby-Computer Fan. Über 450 Seiten **DM 29,80**

26

Mikroprozessor Teil 2, H. Bernstein

Die Fortsetzung unseres ersten so erfolgreichen Buches über Mikroprozessoren. Technologie von Mikroprozessor- und Speicherbausteinen. Festwertspeicher, PROM, REPROM, FIFO, Schieberegister, MPR-Register, ARL-Register, SAR-Register. Aufbau eines Mikroprozessorsystems mit 8080, RAM- und ROM Schnittstellen. Befehlssatz 8080. Über 120 Seiten. **DM 19,80**

27

Mikroprozessor Software Handbuch MSH, C. Lorenz

Grundlagen und Einführung in die Mikroprogrammierung. Grundlagen und Einführung in die wichtigsten Programmiersprachen (BASIC, FORTRAN, ASSEMBLER-Sprachen) Zusammenstellung der wichtigsten Befehlslisten: 8080, Z80, M 6800, National, Fairchild, etc.
Ein Software Handbuch für jeden der mit Mikroprozessoren oder Mikrocomputern zu tun hat. Über 200 Seiten. **Preis DM 29,80**

28

Lexikon und Wörterbuch für Elektronik und Mikroprozessortechnik, LEM, C. Lorenz

Ein Hilfs- und Arbeitsbuch für jeden der sich heute mit der modernsten Elektronik beschäftigt. Viele engl. Ausdrücke werden heute in der Elektronik, Computer- und Mikroprozessortechnik verwendet und oft fehlt uns eine genaueste und präzise Erklärung. Dieses Buch übersetzt Ihnen den englischen Fachausdruck und gibt Ihnen zusätzlich noch eine deutsche Erläuterung und Erklärung dieses Begriffes und was es damit auf sich hat. Ein Lexikon und Wörterbuch in einem einzigen Buch vereinigt. Das Buch, das Sie schon lange gesucht haben. Ca. 250 Seiten. Lieferbar Herbst 1978 **Preis DM 29,80**

29

Mikrocomputer Datenbuch, C. Lorenz

Zusammenstellung der wichtigsten Mikroprozessordaten.

Eine übersichtliche und sehr informative Zusammenstellung der wichtigsten Mikroprozessorbausteine auf dem Markt. 8080A, 8085, 8048, Z80, Z8, 6500, 6800, 2650, 1802, F8, 3870, SC/MP, PACE u. v. a. Daten, Anschlußbilder, wichtige technische und elektrische Daten, Architektur, grundlegende Eigenschaften. Zu jedem Mikroprozessor werden dann auch noch die peripheren Bausteine sowie RAM und ROM Elemente behandelt. Das ideale Handbuch für jeden modernen Elektroniker. Lieferbar Ende 78 **DM 49,–**

30

Aktivtraining - Mikrocomputer (Sammelordne

Ein Mikrocomputerkurs in Form einer Lehrbriefsammlung.

Das Aktivtraining-Seminar erscheint in einzelnen Lehrbriefen, die in einem formschönen praktischen Plastik-Sammelordner geliefert werden. Die Grundausstattung besteht aus 10 Lehrbriefen und reicht von den elementaren Grundlagen bis hin zu praktischen Beispielen. Weiterhin ist ausführliches Mikrocomputer Datenmaterial im Ordner enthalten. Die Lehrbriefe werden regelmäßig erscheinen und können jederzeit nachbezogen werden. Lieferbar Ende 78 **DM 49,–**

31

57 Programme in BASIC, C. Lorenz

Ein Buch mit technisch-wissenschaftlichen Programmen und einer großen Anzahl von Spielprogrammen in BASIC. (Games) Ein Buch für jeden, der sich mit dem faszinierenden Hobby der Mikrocomputertechnik befassen will. Alle Listings sind in BASIC und können auf den meisten Personal Computer Systemen gefahren werden.
Lieferbar Ende 78 **DM 39,–**

32

Circuits Digital Et Pratique, C. Lorenz

Dieses Buch ist eine französische Übersetzung unserer beiden Bücher Nr. 6 und Nr. 13 (IC-Schaltungen und Hobby Elektronik Handbuch HEH)
Ca. 100 Seiten

DM 19,80
FF 39,90

33

Mikrocomputer Programmierbeispiele für 2650, Dr. J. Hatzenbichler

Eine Einführung in die Programmierung von Mikrocomputern an Hand des Prozessors 2650 von Signetics. Viele Programmierbeispiele in Maschinensprache, die Sie auf einem preiswerten Mikroprozessorsystem MIKIT 2650-P2 ausführen können. Zeitschleifenprogramme, Blinkschaltung, Lauflicht, Stufenzähler, Elektronischer Würfel, Stoppuhr, Reaktionszeittester, Computer Musik Programm u. v. a. Jeder Befehl wird genau erläutert und an Hand eines ausführlichen Flußdiagrammes erklärt. Jedes Programm liegt in Form eines Computer-Listings vor. Jedes Programm ist genauestens beschrieben. Sie können so auf einfache Weise die Zusammenhänge erkennen und erwerben damit die Grundkenntnisse zur Erstellung Ihrer eigenen Programme. Sie erkennen, wie man mit nur einer Schaltung (Mikrocomputer) unendlich viele praktische Anwendungsschaltungen realisieren kann. Zu diesem Buche ist auch ein komplett aufgebautes und getestetes Mikrocomputersystem erhältlich, auf dem Sie alle beschriebenen Programme selbst ausführen können. Über 120 Seiten

DM 19,80

105
1

TTL - Experimentierbuch

Eine kleine Einführung in die Digitaltechnik

Grundlagen der Digitaltechnik kurz erklärt. Bauanleitung für einen praktischen Logik-Tester. Viele Experimente und Anwendungsschaltungen mit dem TTL Gatterbaustein 7400. Das ideale Einführungsbuch in die Digitaltechnik.

DM 5,–

106
1

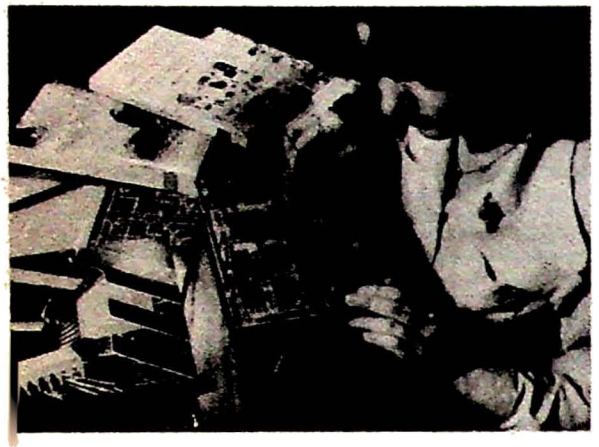
CMOS - Experimentierbuch

Eine kleine Einführung in die CMOS Schaltungstechnik

CMOS Grundlagen, Behandlungshinweise für CMOS Bausteine, Zusammenstellung der wichtigsten CMOS Bausteine und deren Anschlußbilder. Viele praktische CMOS Schaltbeispiele. Ideal für jeden Elektroniker.

DM 5,–

IC KIT Experimentier-platine



Mit der IC-KIT Experimentierplatine wird jetzt das Aufbauen von Schaltungen mit ICs, Transistoren, Widerständen und Kondensatoren zum Kinderspiel. Einfach, schnell und ohne Mühe lassen sich mit dieser hochwertigen Epoxy Versuchsplatine selbst die komplizierten Versuchsschaltungen aufbauen.
Die IC KIT Experimentierplatine Typ WH-1 g eignet sich zum Aufbau von Uhren-, Rechner- und Mikroprozessorschaltungen besonders gut, da sie neben 14,16,24 und 28 poligen DIL-Sockeln auch einen 40 poligen Stecksockel enthält.

Der Einsatz von Adaptern ermöglicht auch das Stecken von diskreten Bauelementen. Weiterhin können Sie sich mit diesen Adaptern selbst kleine Funktionsgruppen aufbauen, die Sie immer wieder verwenden können, zum Beispiel Taktgeneratoren, Logiktester, Tastenentprellschaltungen etc. Der Schaltungsaufbau wird durch das Stecken aller Bauteile und Verbindungsleitungen wesentlich vereinfacht. Die einzelnen Elemente werden nicht beschädigt und bleiben frei von Lötzinn. Sie können immer wieder verwendet werden. Sie sparen Zeit und Geld!

41

Der Bausatz IC KIT WH-1 g enthält: 1 Experimentierplatine in stabiler Epoxy-Ausführung mit den Abmessungen 210 x 150 mm fertig gebohrt und Leiterbahnen verzinnt. Die Bestückungsseite ist mit einem kratzfesten Lack überzogen und bedruckt. Sie können so die gesuchten Anschlüsse beim Experimentieren immer leicht und schnell finden. 340 Messingbuchsen, verschiedenfarbige Stecker mit Querloch (2,6 mm ϕ), Metallfüße und sämtliche Sockel.

Bestell Nr. 41 — Preis DM 79.–

41/1

Adapter zum Stecken von diskreten Bauelementen, 5 Stück in einem Beutel verpackt. Die diskreten Bauteile werden hier einmal angelötet und können dann immer wieder verwendet werden. Einfachste Handhabung.

Bestell Nr. 41/1 — Preis DM 14,90

48

5V Netzgerät stabilisiert für die IC KIT Experimentierplatine Typ WH-1 g. Bausatz komplett mit Netztransformator, Platine, allen Bauelementen, Netzkabel etc. Maximale Welligkeit 1 %, TK = 0,5 mV pro °C, Strombegrenzung. Maximaler Ausgangsstrom 650 mA. Ideal für alle TTL und CMOS Experimente.
Bestell Nr. 48 — Preis DM 39.–

Neue Bücher über Mikro-prozessoren

Programming Microprocessors

Zahlensysteme, Flußdiagramme, Adressierungsarten, Assemblersprachen, Programmierhilfen u. v. a. Das ideale Microprozessorbuch mit Schwerpunkt Software. Über 270 Seiten.

Best. Nr. 985 DM 35,–

Microprocessor/Microprogramming Handbook.

Das praktische Handbuch für jeden der sich mit Mikroproz. beschäftigen will. Über 290 Seiten.

Best. Nr. 785 DM 35,–

Beginners Guide to Computer-Programming.

Die ideale Einführung in die Computerprogrammierung. Gut geeignet für das Selbststudium. Über 480 Seiten.

Best. Nr. 574 DM 39,–

Computer Programming Handbook

Eine genaue Einführung in die Programmierung und EDV mit ausgeführten Beispielen. Der ideale "Ein Buch Computerkurs". Über 510 Seiten.

Best. Nr. 752 DM 45,–

Modern Guide to Digital Logic Processors, Memories, Interfaces.

Speicherschaltungen, Interfaceschaltungen, ECL, Schottky TTL u. v. a. mehr. Über 290 Seiten.

Best. Nr. 709 DM 35,–

Master Handbook of 1001 practical Electronic Circuits.

Ein Schaltungsbuch mit über 1000 Schaltbeispielen mit kurzen Beschreibungen. Über 600 Seiten.

Best. Nr. 800 DM 49,–

Best. Nr.	Stck.	BUCHTITEL BEZEICHNUNG	a DM	ges. DM
1		TBB – Handbuch Band 1	19,80	
2		TBB – Handbuch Band 2	19,80	
3		Elektronik im Auto	9,80	
4		IC – Handbuch, TTL, CMOS, Lin.	19,80	
5		IC – Datenbuch, TTL, CMOS, Lin.	9,80	
6		IC – Schaltungen, TTL, CMOS	9,80	
7		Elektronik Schaltungen	5,–	
8		IC – Bauanleitungshandbuch	19,80	
9		Feldeffekttransistoren	5,–	
10		Elektronik und Radio 4. Auflage	19,80	
11		IC – NF – Verstärker	9,80	
12		Beispiele integrierter Schaltungen	19,80	
13		HEH, Hobby Elektronik Handbuch	9,80	
14		IC – Vergleichsliste	9,80	
15		Optoelektronik Handbuch	19,80	
16		CMOS Teil 1, Einführung	19,80	
17		CMOS Teil 2, Schaltbeispiele	19,80	
18		CMOS Teil 3, Schaltbeispiele	19,80	
19		IC – Experimentier Handbuch	19,80	
20		Operationsverstärker	19,80	
21		Digitaltechnik Grundkurs	19,80	
22		Mikroprozessorn 2. Auflage	19,80	
23		Elektronik Grundkurs	9,80	
24		Mikrocomputer Technik	29,80	
25		Hobby Computer Handbuch	29,80	
26		Mikroprozessor Teil 2	19,80	
27		Mikroprozessor Software Handbuch	29,80	
28		Lexikon + Wörterbuch	29,80	
29		Microcomputer Datenbuch	49,80	
30		Aktivtraining Microcomputer	49,80	
31		57 Programme in BASIC	39,–	
32		Circuits Digital Et Pratique	19,80	
33		Microcomputer Programmierbeispiele	19,80	
41		IC-KIT Experimentierplatine WH-1 g	79,–	
100/1		TTL-Experimentierbuch	5,–	
100/1		CMOS-Experimentierbuch	5,–	

Kurzlehrgang TTL CMOS

21/41

TTL – Kurzlehrgang
Der ideale Einführungskurs für das Selbststudium, den Unterricht, die Ausbildung und für den Amateur. Das Buch Nr. 21 zusammen mit unserem IC-KIT Experimentierplatinenbausatz WH-1g bilden die ideale Grundlage für diese TTL-Einführung. Kreuzen Sie auf der Bestellkarte einfach die Bestellnummern 21 und 41 an, Sie erhalten dann umgehend diesen wertvollen Kurzlehrgang zugeschickt.
Best. Nr. 21/41 DM 98,80

CMOS – Kurzlehrgang
Der ideale Einführungskurs in die moderne C-MOS Technik. Grundlagen, Entwurf, Versuchsschaltungen. Das Buch Nr. 16 zusammen mit der IC-KIT Experimentierplatine WH-1g bilden die ideale Grundlage für diesen interessanten Kurs. Kreuzen Sie auf der Bestellkarte einfach die Bestellnummern 16 und 41 an. Sie erhalten dann umgehend diesen C MOS-Kurs.
Best. Nr. 16/41 DM 98,80

16/41

Abs.
Name

Straße:

Ort:

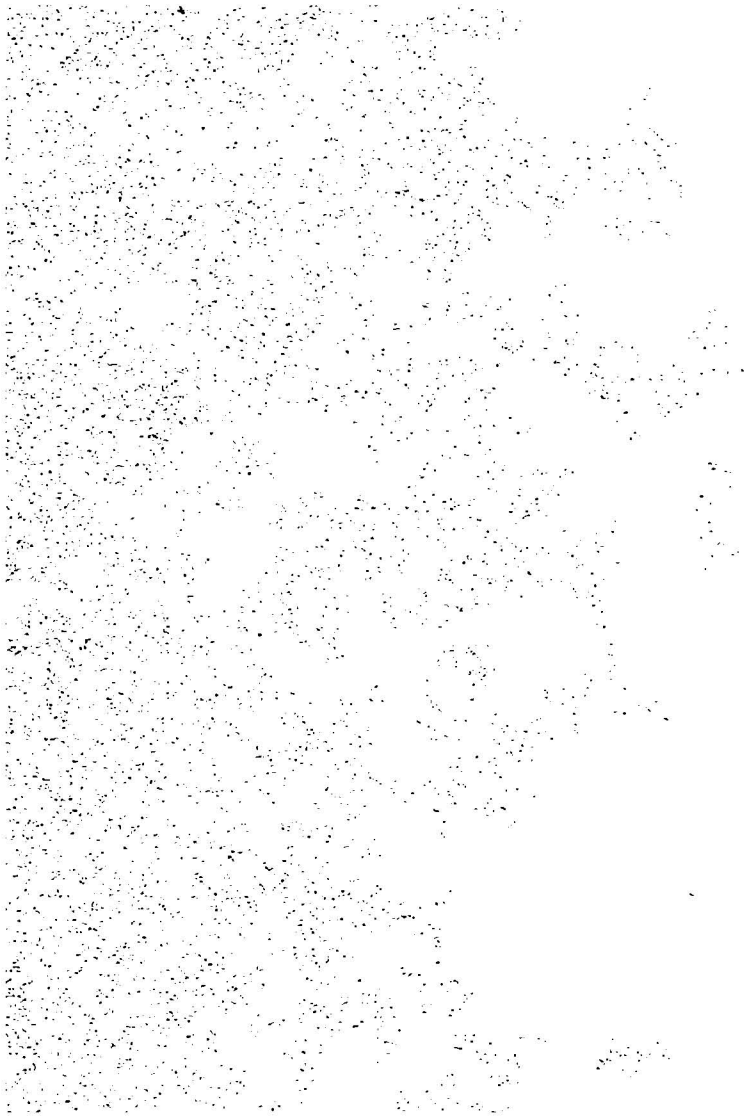
Bestellkarte

Den Gesamtbetrag für die umseitig bestellten Bücher bitte ich per Nachnahme zu berechnen/habe ich heute auf Ihr Postscheckkonto München 15994-807 überwiesen. Nichtzutreffendes bitte streichen.

○ Bitte liefern Sie mir auch die Experimentierplatine IC - KIT WH-1 g für 14,16,24,28 und 40 polige DIL IC-Gehäuse. Bausatz komplett: DM 79.–

○ Bitte liefern Sie von jedem neu erscheinenden Buch ein Muster per Nachnahme. (Rückgaberecht)

Ing.
W. Hofacker GmbH
Verlag
Postfach 437
8000 München 75

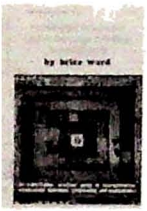


MICROPROZESSOREN MICROCOMPUTER

Ihr Fachverlag für aktuelle Elektronik

Best.-Nr. 22

Best.-Nr. N8



Best.-Nr. 785

Best.-Nr. 985

Bestell-Nr.	Titel	Preis
1	Transistor Berechnungs- u. Bauanl. HB 1, 128 Seiten	19,80
2	Transistor Berechnungs- u. Bauanl. HB 2, 139 Seiten	19,80
3	Elektronik im Auto, 50 Seiten	9,80
4	IC-Handbuch, TTL, C MOS, Linear, 130 Seiten	19,80
5	IC-Datenbuch, TTL, C MOS, Linear, 115 Seiten	9,80
6	IC-Schaltungen, TTL, C MOS, Linear, 38 Seiten	9,80
7	Elektronik-Schaltungen, 65 Seiten	5,--
8	IC-Bauanleitungs-Handbuch, 125 Seiten	19,80
9	Feldeffekttransistoren, 45 Seiten	5,--
10	Elektronik und Radio, 40 Seiten	5,--
11	IC-NF-Verstärker, 65 Seiten	9,80
12	Beispiele integrierter Schaltungen (BIS), 130 Seiten	19,80
13	HEH, Hobby Elektronik Handbuch, 55 Seiten	9,80
14	IC-Vergleichsliste, 50 Seiten	9,80
15	Optoelektronik-Handbuch, 106 Seiten	19,80
16	C MOS, Teil 1, Einführung, Entwurf, Schaltbeispiele, 140 Seiten	19,80
17	C MOS, Teil 2, Entwurf und Schaltbeispiele, 140 Seiten	19,80
18	C MOS, Teil 3, Entwurf und Schaltbeispiele, 140 Seiten	19,80
19	IC-Experimentier-Handbuch, 120 Seiten	19,80
20	Operationsverstärker	19,80
21	Digitaltechnik Grundkurs, 130 Seiten	19,80
22	Mikroprozessoren, Eigenschaften und Aufbau, 120 Seiten	19,80
23	Elektronik Grundkurs, Kurzlehrgang Elektronik, 150 Seiten	9,80
24	Mikrocomputer-Anwender-HB, MAH, 200 Seiten	29,80
25	Hobby Computer Handbuch HCH, 150 Seiten	29,80
26	Mikroprozessor, Teil 2, 120 Seiten	19,80
N 8	SC/MP Programm. + Ass HB.	19,80

Bestell-Nr.	Titel **Bücher in englischer Sprache**	Preis
800	1001 Master Handbook über 600 Seiten	49,--
785	Microprocessor/Microprogramming über 290 Seiten	35,--
985	Programming Microprocessors 280 Seiten	35,--
709	Modern Guide to Digital Logic, 290 Seiten	35,--
574	Beginners' Guide to Comp Progr über 480 Seiten	39,--
874	Master Handbook of Digital Logic, 380 Seiten	45,--
774	Digital/Logic Electronics HB über 300 Seiten	35,--
828	Switching Regulators, 253 Seiten	24,80

Universal Experimentierplatine IC-KIT Typ WH-1g

Für ICs im 40-, 28-, 24-, 16- und 14poligen DIL-Gehäuse. Abmessungen 210 x 150 mm. Stab. Epoxy-Ausführung. Ideal für alle Versuchsschaltungen mit ICs und diskreten Bauelementen. Kein Löten mehr. Alle Verbindungen und Bauteile werden gesteckt. Sie sparen Zeit und Geld, da alle Teile frei von Lötzinn bleiben und immer wieder verwendet werden können. Bausatz enthält alle Teile incl. Sockel.
Best.-Nr. 41 DM 79,-

ING. W. HOFACKER VERLAG

Tegernseestraße 18
815 Holzkirchen / Obb.
Tel. 08024 / 73 31

Verlagsprogramm

Übersicht lieferbarer Bücher

Bestell Nr.	ISBN	Verfasser	Titel	Preis DM
1	3-921682-01-0	Hofacker	Transistor Berechnungs- und Bauanleitungshandbuch, Band 1	19,80
2	3-921682-02-9	Hofacker	Transistor Berechnungs- und Bauanleitungshandbuch, Band 2	19,80
3	3-921682-03-7	Gebauer	Elektronik im Auto	9,80
4	3-921682-04-5	Lorenz	IC-Handbuch, TTL, CMOS, Linear	19,80
5	3-921682-05-3	Steinbach	IC-Datenbuch, TTL, CMOS, Linear	9,80
6	3-921682-06-1	Steinbach	IC-Schaltungen, TTL, CMOS, Linear	9,80
7	3-921682-33-9	Hofacker	Elektronik Schaltungen	5,–
8	3-921682-08-8	Lorenz	IC-Bauanleitungs-Handbuch	19,80
9	3-921682-09-6	Lorenz	Feldeffekttransistoren	5,–
10	3-921682-34-7	Lorenz	Elektronik und Radio	5,–
11	3-921682-11-7	Lorenz	IC-NF Verstärker	9,80
12	3-921682-12-6	Bernstein	Beispiele Integrierter Schaltungen (BIS)	19,80
13	3-921682-13-4	Lorenz	HEH, Hobby Elektronik Handbuch	9,80
14	3-921682-14-2	Lorenz	IC-Vergleichsliste	9,80
15	3-921682-15-0	Lorenz	Optoelektronik Handbuch	19,80
16	3-921682-16-9	Bernstein	CMOS Teil 1 Einführung Entwurf, Schaltbeispiele	19,80 19,80
17	3-921682-17-7	Bernstein	CMOS Teil 2 Entwurf und Schaltbeispiele	19,80
18	3-921682-18-5	Bernstein	CMOS Teil 3 Entwurf und Schaltbeispiele	19,80
19	3-921682-19-3	Lorenz	IC-Experimentier Handbuch	19,80
20	3-921682-20-7	Lorenz	Operationsverstärker	19,80
21	3-921682-21-5	Lorenz	Digitaltechnik Grundkurs	19,80
22	3-921682-22-3	Bernstein	Mikroprozessoren, Eigenschaften und Aufbau	19,80
23	3-921682-23-1	Lorenz	Elektronik Grundkurs Kurzlehrgang Elektronik	9,80
24	3-921682-35-5	Hans Peter Blomeyer-Partenstein	Mikrocomputer-Anwender-Handbuch MAH	29,80
25	3-921682-25-8	C. Lorenz	Hobby Computer Handbuch HCH	29,80
26	3-921682-26-8	H. Bernstein	Mikroprozessor Teil 2	19,80
27	3-921682-27-4	C. Lorenz	Mikrocomputer Software Handbuch MSH	29,80
28	3-921682-28-2	C. Lorenz	Lexikon + Wörterbuch für Elektronik und Mikroprozessortechnik LEM	29,80
29	3-921682-29-0	H. Bernstein	Handbuch für theoretische und angewandte Elektronik Teil 1	49,80
30	3-921682-30-4	H. Bernstein	Handbuch für theoretische und angewandte Elektronik Teil 2	
31	3-921682-31-2	C. Lorenz	Basic Digital Electronics Course (en	
31	3-921682-32-0	C. Lorenz	Circuits Digital Et Pratique (französ	
41			Experimentierplatine für 14, 16, 24, pol. DIL ICs.	

Ing. W. Hofacker GmbH Ver

8 München 75